Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Prof. Dr.-Ing. P. Tran-Gia

Modeling Crowdsourcing Platforms - A Use-Case Driven Approach

Matthias Johannes Wilhem Hirth

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 02/16

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr. Ing. P. Tran-Gia
Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Am Hubland
D-97074 Würzburg

Tel.: +49-931-31-86630
Fax.: +49-931-31-86632
email: trangia@informatik.uni-wuerzburg.de

Satz

Reproduktionsfähige Vorlage vom Autor.
Gesetzt in LaTeX Linux Libertine 10pt.

ISSN 1432-8801

Modeling Crowdsourcing Platforms - A Use-Case Driven Approach

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius–Maximilians–Universität Würzburg

vorgelegt von

Matthias Johannes Wilhem Hirth

aus

Würzburg

Würzburg 2016

Eingereicht am: 20.05.2016

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr.-Ing P. Tran-Gia

2. Gutachter: Prof. Dr.-Ing. Klaus Diepold

Tag der mündlichen Prüfung: 09.08.2016

Danksagung

In den vergangenen Jahren während denen diese Dissertation entstanden ist, habe ich Unterstützung von einer Vielzahl von Personen erhalten. Sie haben damit einen wesentlichen Teil dazu beigetragen, dass ich diese Arbeit erfolgreich abschließen konnte.

Allen voran gilt mein Dank meinen Doktorvater und Betreuer Prof. Dr.-Ing. Phuoc Tran-Gia. Er hat mir die Change gegeben in einem noch relativ jungen Forschungsgebiet an der Schnittstelle zwischen Mensch und Technologie zu promovieren. Hierbei hat er mich stets ermuntert neue Blickwinkel, auch außerhalb des Fachbereichs der Informatik, zu betrachten und hat meine wissenschaftliche Arbeit in gemeinsamen Diskussionen entscheidend mitgestaltet. Ferner ermöglichte mir Prof. Dr.-Ing. Phuoc Tran-Gia im Rahmen von zahlreichen Konferenzen, Fachvorträgen und Kooperationen den Austausch und die Zusammenarbeit mit in- und ausländischen Kollegen aus Wissenschaft und Industrie. Dieser Erfahrungsaustausch förderte die Entstehung einer Vielzahl von Ideen die in meine Dissertation mit eingeflossen sind. Besonders bedanke ich mich auch für das von Prof. Dr.-Ing. Phuoc Tran-Gia entgegengebrachte Vertrauen, dass ich eigenverantwortlich Projekte definieren und durchzuführen durfte. Dies ermöglichte mir mich in meiner Fachrichtung, als auch darüber hinaus weiterzubilden und wichtige Erfahrungen zu sammeln. Als einen stets besonders wichtigen Aspekt bei meiner Arbeit habe ich das Arbeitsumfeld gesehen. Hier hat es Prof. Dr.-Ing. Phuoc Tran-Gia in besonderer Weise geschafft ein positives Umfeld am Lehrstuhl zu schaffen.

Weiterhin bedanke ich mich bei Prof. Dr.-Ing. Klaus Diepold für die Zusammenarbeit in den vergangenen Jahren, die angeregten Diskussionen in Dagstuhl und besonders dafür, dass er als Zweitgutacher zur Verfügung stand. Ferner möchte ich mich bei Prof. Dr. Birgit Lugin und Prof. Dr. Samuel Kounev bedanken, dass sie sich kurzfristig und trotz Urlaubszeit bereit erklärt haben, als Prüfer bei meiner Disputation zu fungieren. I also thank Nhatvi Nguyen for giving me the unique possibility to gain insights in the commercial aspects of crowdsourcing. He supported many of the research studies that led to this dissertation. Des Weiteren möchte ich mich bei Dr. Christoph Rensing, Steffen Schnitzer, Prof. Dr. Wehnes und Peter Karg für die langjährige gute Zusammenarbeit in den gemeinsamen Forschungsprojekten bedanken. I am also grateful that I met Deniz Iren during my research activities. His dedication to research was and is always an inspiration.

Diese Arbeit wäre auch ohne die gute Atmosphäre zwischen meinen aktuellen und ehemaligen die Kolleginnen und Kollegen Kathrin Borchert, Valentin Burger, Lam Dinh-Xuan, Dr. Michael Duelli, Steffen Gebert, Stefen Geißler, Nicholas Gray, Dr. Matthias Hartmann, Dr. Henjes, Dr. David Hock, Prof. Dr. Tobias Hoßfeld, Dr. Michael Jarschel, Dr. Dominik Klein, Dr. Frank Lehrieder, Stanislav Lange, Prof. Dr. Michael Menth, Christopher Metter, Anh-Ngoc Nguyen, Dr. Simon Oechsner, Dr. Rastin Pries, Dr. Daniel Schlosser, Dr. Christian Schwartz, Michael Seufert, Prof. Dr. Barbara Staehle, Prof. Dr. Dirk Staehle, Dr. Florian Wamser und Dr. Thomas Zinner nicht in dieser Form möglich gewesen. fand man immer ein offenes Ohr, sowohl bei fachlichen Fragen oder schlicht wenn die Technik einmal gestreikt hat. Besonders möchte ich hier Prof. Dr. Hoßfeld nennen, der mir jahrelang als Gruppenleiter mit Rat und Tat zur Seite stand und der meine bisherige wissenschaftliche Arbeit stark geprägt hat. Ebenso möchte ich hier Prof. Dr. Barbara Staehle erwähnen. Sie hat mich als Betreuerin meiner Diplomarbeit zu Beginn meiner Tätigkeit am Lehrstuhl fachlich geleitet und gab mir nun wertvolle Kommentare bei der Anfertigung der Dissertation. Ein weiterer zentraler Anlaufpunkt in fachlichen Fragen und Diskussionen ist in den letzten Jahren Dr. Thomas Zinner für mich geworden, wofür ich mich bedanken möchte.

Der guten Atmosphäre am Lehrstuhl ist es auch zu verdanken, dass die gemeinsamen Aktivitäten oft nicht mit dem Arbeitstag endeten. So fanden sich immer wieder Gelegenheiten für gemeinsame Programmierprojekte mit Dr. Christian Schwartz, oder Abenteuer in digitalen Welten mit Valentin Burger, Dr. Michael Jarschel und Dr. Christian Schwartz. Ebenso kamen auch sportliche Aktivitäten wie etwa gemeinsames Klettern mit Valentin Burger, Dr. Matthias Hartmann, Stanislav Lange, Dr. Frank Lehrieder und Prof. Dr. Barbara, sowie Sparring beim Judo Training mit Prof. Dr. Michael Menth und Dr. Daniel Schlosser oft nicht zu kurz. Nicht unerwähnt bleiben sollen hier natürlich auch die Skiausflüge und die vielen gemeinsamen Abende mit all den anderen Lehrstuhlmitgliedern. Vielen Dank für die vielen schönen Stunden.

Weiterhin bedanke ich mich bei unserer Sekretärin Alison Wichmann für die unentwegte Unterstützung in Verwaltungsangelegenheiten und bei der Hiwiverwaltung. Ein Dankeschön gebührt natürlich auch den damaligen Studentinnen und Studenten Philipp Amrehn, Armin Beutel, Kathrin Borchert, Nicholas Kuhn, Veronika Lesch, Stephan Oberste-Vorth, Anh-Ngoc Phung, Johann Scherer, Sven Scheuring, Martin Scholz, Michael Seufert, Susanna Schwarzmann die in den vergangenen Jahren bei mir ihrer Abschlussarbeit angefertigt haben, sowie allen Hiwis die mich unter Anderem bei der Projektarbeit und der Lehre unterstützt haben.

Schließlich möchte ich mich ganz besonders bei meinen Freunden und Familie bedanken. Allen voran meinen Eltern Gisela und Werner Hirth. Sie haben mir während der Schule-, Universitäts- und Promotionszeit alle nur erdenkliche Unterstützung geboten die notwendig war diese erfolgreich abzuschließen. Abschließend möchte ich mich bei meiner Partnerin Franziska Winkler bedanken. Obwohl sie mich in den vergangenen Jahren oft auch abends und an Wochenenden mit meiner Arbeit hat teilen müusste, bestärke sie mich doch stets in meinen Zielen und gab mir den notwendigen Rückhalt.

Contents

1 Introduction

Computer systems are becoming more and more powerful and capable of assisting or replacing human labor force in many areas of everyday life. Today's state of the art systems are able to control industrial plants, drive vehicles, or process enormous amounts of data as shown by companies like Google or Facebook. The capabilities of these systems are continuously improving with new developments in the hardware sector, with chipsets getting smaller and faster, or by further improvements on the software side like deep learning. This progress fosters applications that have not been possible a few year ago, e.g., the virtual assistants on smart devices or real-time translations of voice chats.

However, there are still tasks even the most advanced computer systems are not able to solve efficiently, yet, even if the tasks are easily solved by humans. This includes tasks like categorization or annotation of image and summarization of texts. The same applies to any task that includes subjective or emotional ratings, like assessing the appeal of images, music, or a scenery. In traditional forms of work organization, full-time employees are hired to complete tasks that cannot be automated. However, this is usually not cost-effective for simples tasks like the ones just mentioned.

One possibility to solve such tasks effectively is using crowdsourcing. The term crowdsourcing was created by Jeff Howe in 2006, who defined crowdsourcing as *[...] the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call* [35]. Compared to traditional forms of work organization, crowdsourcing work does not require a long-term relationship between the employer offering the task and the employee or worker completing the tasks. Further, a task is usually not assigned to a dedicated worker, but the workers choose freely which task to work on.

Still the question remains, how to efficiently form the required crowd for the crowdsourcing approach. Here, the increasing availability of Internet access via computers and smart devices can be of help. Today, the Internet connects people all over the globe, who form a potentially huge and highly available workforce. New web-bases services emerged, so called crowdsourcing platforms, that try to make this work-force accessible.

Already today, commercial crowdsourcing platforms need to cope with technical challenges. The user base often consists of hundreds of thousands users that complete millions of tasks. This results in large amounts of data that need to highly available all over the globe, due to the distributed user base. Additionally, more and more services emerge that use crowdsourcing for real-time, or almost real-time data processing, e.g., for content filtering in web applications. The technical requirements on the infrastructure of the platforms will increase even further in the future, due to the constant growth of the platforms and the tighter coupling of machine clouds and crowdsourcing platforms.

Services using a combination of crowdsourcing and machine clouds can benefit from the strengths of both approaches. Machine clouds offer a cost effective way to process large amounts of data, if the processing can be done algorithmically. On the other side, crowdsourcing offers easy access to a human workforce, which can process data points that cannot be handled by the machine cloud. A typical example for such a combined human-machine application is text digitalization. In an initial step, a digital image of a text is processed using optical character recognition. The result is then refined and proof-read by human editors to increase the accuracy of the digitalized text. Services combining machine clouds and crowdsourcing are often used for data processing tasks that require high accurate results. While the accuracy of software solutions can often be easily assessed, assessing the accuracy of crowdsourcing results is challenging. This is mainly caused by the fact that the quality of crowdsourcing results can vary significantly due to the human contributors. Different methods have been proposed to increase the quality of crowdsourcing results, e.g., posting redundant tasks and aggregating the workers' results. However, the set of applicable methods highly depends on the specific crowdsourcing use case.

This monograph tackles the challenges of current crowdsourcing systems in the following way. Our first goal it to gain an understanding of existing crowdsourcing systems. This knowledge is then used to develop analytical models for dimensioning crowdsourcing platform infrastructure and to model the behavior of the platform users. The second goal is to optimize existing crowdsourcing workflows and systems, where we especially focus on assessing and increasing the quality of task results. To this end we analytically evaluate existing methods and also propose novel mechanisms. The last goal of this monograph is to illustrate the benefits and challenges of crowdsourcing for research using two complementary use cases. We illustrate these use cases with results from numerous crowdsourcing studies and derive general best practices that can be applied to a wide range of crowdsourcing tasks. A detailed summary of the scientific contribution of this monograph is given in the following.

1.1 Scientific Contribution

Figure 1.1 visualises the studies carried out during this thesis and categorizes them according to the utilized methodology depicted on the x-axis and the focus shown on the y-axis. The methodologies include measurements, user studies, mathematical analyses, simulations, and the design of new use-cases and mechanisms. The focus of the studies can be distinguished in Quality of Experience (QoE) and Quality of Service (QoS), Online Social Networks (OSNs) and crowdsourcing. This monograph focuses on the contributions in the field of crowdsourcing, the publications used in this monograph are marked with the notion $x^{(y)}$, which indicates that the publication x is discussed in Chapter y.

The first major contribution is the development of generalizable models for the growth of crowdsourcing platforms and the activity of platform users. The users of a crowdsourcing platform and the underlying infrastructure are cou-

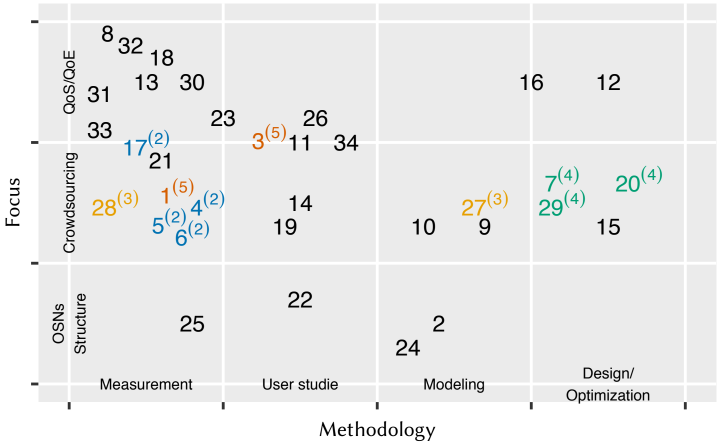


Figure 1.1: Cartography of Contributions

pled more tightly than in most other network-based services as the users are an integral part of the system. Therefore, an appropriate dimensioning of the technical infrastructure as well as the human workforce is crucial for a successful operation of a crowdsourcing system. To achieve this, a good understanding of the system and the involved actors is required and models need to be developed that enable the analysis of different possible system configurations. To this end, we review existing commercial platforms and develop a simple classification scheme for crowdsourcing tasks based on their complexity that can be widely applied. Additionally, we also develop a classification scheme for crowdsourcing platforms. Both classification schemes allow a generalization from specific tasks and platforms to task and platform categories with common attributes which is crucial for modeling those complex systems. Further, we provide insights into a commercial crowdsourcing platform to retrieve information about the geographical distribution of the users and how the platform is used by the different types submitting work to the platform and completing work on the platform.

The second contribution concerns the optimization of quality assurance mechanisms for crowdsourcing. Quality assurance is one of the main factors for the successful application of crowdsourcing and crucial due to the varying result quality form the anonymous workers. We demonstrate that the workers' interactions with the task interface can be used as an indicator of the quality of the workers' results. In contrast to existing methods using gold standard data, this approach does not impose additional workload to the workers and thus helps to reduce task costs. As a main contribution, we introduce analytical models for assurance mechanisms that can help to assess the suitability of the mechanisms for a wide range of tasks. Both, the costs and the resulting quality are taken into account.

The third contribution comprises best practices for using crowdsourcing for network measurement and subjective studies. In a first step, we review existing research methodologies for both fields and identify weaknesses that can be overcome with crowdsourcing. Using the example of representative use cases, we show that methodologies and test setups used in laboratory settings cannot be applied directly to the crowdsourcing environment. To overcome this, we develop guidelines, like the two-stage design for the recruiting process of test participants, for crowdsourced studies and show their feasibility. The requirements of crowdsourced network measurements and crowdsourced subjective studies are also representative for a large number of crowdsourcing tasks, which makes the developed best practices applicable for a wide rage of tasks.

1.2 Outline

The reminder of this monograph is structured as follows. Chapter 2 gives an general introduction in the field of crowdsourcing and presents a categorization approach for both, crowdsourcing tasks and crowdsourcing platforms. Further, it discusses technical challenges for crowdsourcing platforms and conceptual challenges of the crowdsourcing paradigm.

Chapter 3 aims at gaining a better understanding of crowdsourcing platforms and their users. To this end, we first analyse the users of a commercial crowdsourcing platform with respect to their socio-economic background. Thereafter, we have a closer look on how they use the platform. Based on the results from the data analysis we evaluate different growth models and assess their suitability for predicting the future development of the platform's user base. Finally, we develop a fluid model to describe the activity of users on a platform.

In Chapter 4, we evaluate and optimize methods to increase the quality of results obtained from crowdsourcing. We first detail on an approach to assess the quality of an individual worker. For that, we monitor the interactions of the worker with the task interface and predict the quality of the task results based on the coherence of the monitoring data with an expected interaction pattern. Second, we introduce cost and accuracy models for quality assurance mechanisms that are based on the aggregation of multiple submissions from different workers. Here, we show that both considered mechanism result in the same quality of results but at considerably different costs.

Chapter 5 presents best practices for crowdsourcing studies using two different use cases, crowdsourcing based network measurements and subjective studies for evaluating video quality. Both use cases illustrate orthogonal applications. Most network measurements can be realized as automated tests that run on workers devices and do not require a lot of interactions with the workers themselves. The results are mainly observations of objective technical parameters. In contrast, users studies aim explicitly collecting subjective feedback from the workers, which can differ significantly depending on individual preferences. For both fields, we show the benefits of performing additional crowdsourcing-based tests, was well as the limitations of such tests. Additionally, common pitfalls are illustrated and methods and techniques are presented on how to overcome those.

Chapter 6 summarizes the presented results and achievements of this monograph.

2 Crowdsourcing Platforms: Background and Challenges

The Internet has changed the way people communicate and interact with each other like almost no other invention before. With the raise of new communication services like Skype, Voice over IP, or other messenger solutions it is easily possible to connect with people all over the world almost instantly and at extreme low costs. In conjunction with the increasing number of mobile devices, the world becomes globally connected. As a consequence of this tight connection, it becomes increasingly easy for people to discuss, exchange ideas, and collaborate on a national or international scale. This in turn can result in large scale community projects like Wikipedia or new service platforms like Online Social Networks (OSNs).

However, these new communication possibilities also change the way work places look like and how work is organized. Modern communication systems allow engineers to maintain industry plants on other continents, surgeons can treat people hundreds of kilometers away, and flexible home office solutions became part of everyday life in small to large enterprise. Additionally to bringing traditional work "online" the Internet also fosters the development of new forms of work organization, like global freelancer market places or crowdsourcing.

The Internet can be considered as the key driver for crowdsourcing as a new form of work organization that differs significantly from traditional "off-line" forms of work organization. Since crowdsourcing is becoming more and more popular, numerous commercial applications recently evolved. While all of them follow a basic common scheme, the platform types can be distinguished by the type of work they focus on and their customers. Crowdsourcing platforms today already accumulate hundred thousands of people, making them possible future Internet traffic hot-spots similar to OSNs today. This and the tightening of the interconnection of crowdsourcing platforms and machine clouds might also imposes new challenges on the Internet infrastructure in the future.

Considering this, the remainder of this chapter is structured as follows. First, we give a general introduction into crowdsourcing in Section 2.1. This section shows how crowdsourcing fits into the evolution of work organization and introduces classification schemes for both, offered work and commercial crowdsourcing platforms. Section 2.2 discusses the challenges arising from crowdsourcing platforms for the Internet infrastructure and the challenges arising from the connection of crowdsourcing platforms with cloud services. Finally, Section 2.3 gives an overview of current challenges of crowdsourcing platforms. The content of this chapter is mainly taken from [4–6, 27, 29].

2.1 Crowdsourcing and the Granularity of Work

Crowdsourcing can be viewed as a further development of the traditional outsourcing principle, by reducing the administrative overhead and the size of the outsourced task. In the first part of this section, we therefore show how the organization of work and the granularity of work evolved. Thereafter, we detail on the basic principle how crowdsourcing is currently realized and present categorization approaches for both, Crowdsourced work and crowdsourcing services.

2.1.1 Evolution of Work Organization

The way work was organized and granularity of the work units has significantly changed during the 10 to 15 years, leading to new concepts like crowdsourcing. This evolution is illustrated in Figure 2.1.

The largest work package we consider is a *project*, e.g. a web application. In traditional forms of work organization, the *employer* delegates a project to a

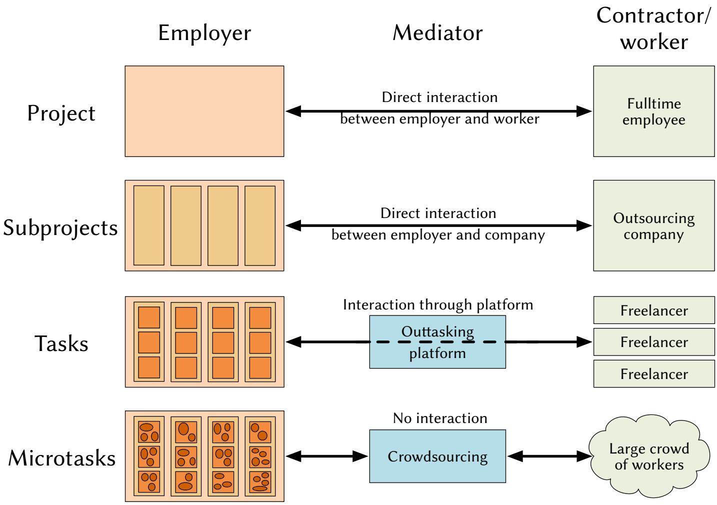


Figure 2.1: Evolution of work organization and granularity of work [6].

designated *employee* or a group of employees. The employee or group of employees completes the whole project in a given amount of time, which can be anything between a few weeks up to several years. In this case, the employer and employees have usually a long term contract and the employee receives a predefined remuneration.

To speed up this process or due to the lack of expertise of the available company employees, parts of the project, so called *sub-projects* might be completed by an external company. Considering the web application example, sub projects might be the front- and the backend of the application. In this case, the employer chooses directly which firm should accomplish the project and negotiates with this firm the terms of the contract.

The next smaller granularity is a *task*. This can be viewed as a small part of the project where no knowledge about the whole project is required, like the design of the landing page of the web application. In contrast to tasks, subtasks can be accomplished by individuals, e.g., freelancers. In order to access the freelancers, the employer uses an *out-tasking platform*. On out-tasking platforms, freelancers can upload a personal profile including references, skills and salary expectations. An employer chooses a freelancer according to his demands. In contrast to outsourcing a project or a task, the employer does not directly communicate with the freelancer nor pays him directly. The out-tasking platform acts as a mediator between them and provides the required infrastructure as paid service.

The finest granularity of work is the *microtask*. In our web application example this could be, e.g, the creation of a logo for the web page. A microtask can be accomplished within a few minutes to a few hour and thus does not need a long term employment. Further, it is irrelevant for the employer who accomplishes the task and usually the task has to be repeated several times. The repetitive tasks are combined in a *campaign*, which the employer submits to the crowdsourcing platform. Similar to the out-tasking platform, the *crowdsourcing platform* acts as a mediator between the employer and the anonymous *workers*. However, the workforce in the crowdsourcing approach is not a *designated* worker but a *human cloud* which is completely abstracted from the employer thought the platform. In the following we describe the technical functionality of these platforms. Concrete examples of currently available commercial providers for different types of tasks are given in Section 2.1.3.

2.1.2 The Crowdsourcing Scheme

A common crowdsourcing environment involves three different types of actors: *Employers*, *workers*, and *(platform) providers*. *Employers* and *workers* are subsumed under the term *users*. The *employers* are crowdsourcing users who offer new tasks and seek other crowdsourcing users to complete the offered work. Crowdsourcing *workers* are users who are working on tasks and submit results or proofs that they have completed the task. Unlike to traditional forms of work organization, an employer does not selectively choose a worker for a certain task, but offers the task to a large crowd of workers who can freely choose to work on this task or not. The Crowdsourcing *(platform) provider* provides and maintains the required technical infrastructure for this process. In the remainder of this monograph, we use the terms *microtask* and *task* in the context of crowdsourcing equivalently.

A typical workflow on a commercial crowdsourcing platform comprises the following steps illustrated in Figure 2.2.

1) In a first step, the employer submits a *campaign* to the crowdsourcing platform. A campaign is an entity aggregating similar *tasks*. It includes a description of the task, which task *result* the workers have to submit back to the platform, respectively how the workers have to *proof* a completed task, the reward per task, and how many tasks are needed.

2) A list of available tasks is provided to the workers and the worker can freely choose which task to work on.

3) After task completion, the worker submits the task result or proof specified by the employer in the task description to the platform.

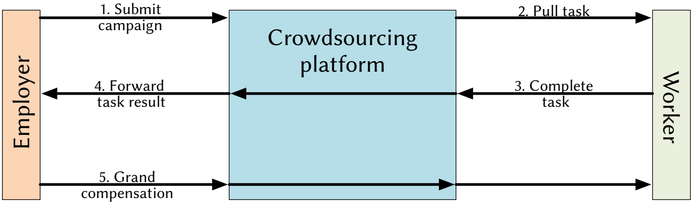


Figure 2.2: Crowdsourcing workflow.

4) The submitted task proofs or results might optionally be pre-processed by the platform and then forwarded to the employer.

5) The employer reviews the submitted results and checks their validity. If a task result is approved by the employer, a compensation is granted to the workers. If a task is not valid, the result is discarded and resubmitted to the crowd. In this case the worker is not rewarded in most cases. The compensation for a task can either be monetary - which is the case in most commercial solutions - or can be non-monetary. Hereby, non-monetary rewards might include public acknowledgements of the contributors, access to collected data, or virtual currencies.

2.1.3 Categorization of Crowdsourcing Tasks

The crowdsourcing scheme can be applied to a multitude of different problems and the resulting crowdsourcing tasks can be categorized according to different and fine granular taxonomies [36, 37]. However, we propose on a more general categorization of tasks and use following three different types of task categories: (1) *Routine Tasks*, (2) *Complex Tasks*, and (3) *Creative Tasks*. A similar categorization was also proposed at the same time by Schenk et al. [38].

Routine tasks are tasks which can be completed with just a few clicks and do not require prior knowledge about the task, a special skill set, or dedicated hardware. Common examples are tagging or classification of images or texts, or the extraction of data from web pages or scanned documents. Besides this, a lot of search engine optimization tasks belong to this category, like the creation of back links or the distribution of content via social media platforms like Facebook or Twitter. An example on how to enable cost optimal quality control for routine tasks is discussed later in Section 4.2. Section 5.1 discusses, how routine tasks can be used to improve current network measurement techniques. Well known mediators for routine tasks are Amazon Mechanical Turk (MTurk) [39] or Microworkers [40].

The second category of tasks, so called *complex tasks*, comprises of tasks like content generation, writing of forum and blog posts, content commenting, writing of product reviews, and the participation in surveys. Also tasks including testing of web applications, web services, and software in order to detect bugs or for improving the application's design, usability, and QoE can be subsumed in this category. In contrast to routine tasks, this set of task requires a limited set of skills and also prior experience related to the task. A content generation task, e.g., requires writing skills and also background knowledge of the topic. These prerequisites can usually not be learned during the execution of the task itself. Further tasks that require special devices belong to this category, like mobile crowdsourcing tasks provided by Streetsptr [41]. Examples for complex tasks and best practices for improving their results are further discussed in Section 5.2.

Creative Tasks, refer to difficult and challenging tasks like software and web development, solving of complex problems or research challenges, the generation of new ideas, or design related tasks. These tasks require high trained skills or a high education level. One example for crowdsourcing platforms focusing on this type of tasks is Innocentive [42]. Innocentive offers companies the possibility to publish challenges related to production processes or research questions, which cannot be solved internally either due to the lack of expertise or missing manpower. External experts can answer to the posted challenges with the submission of possible solutions, which are reviewed by the posting company. The submitter of the approved solution is then rewarded. Another well known

Table 2.1: Microtask categories

	Routine Task	Complex Task	Creative Task
Worker prerequisites	None	Basic skills, specialized hardware	High skills, high education level
Exemplary tasks	Basic categorization tasks, basic SEO tasks	Content creation, mobile crowdsourcing tasks	Research and development, design tasks
Exemplary platform providers	MTurk, Microworkers	Cloudfactory, Streetsptr	Innocentive, 99designs

example is 99designs [43] where a crowd of design affine users develops and realizes designs for multiple purposes, e.g., web pages, logos, and print media.

The main characteristics of routine tasks, complex tasks, and creative tasks, as well as exemplary platforms supporting these types of tasks are summarized in Table 2.1.

2.1.4 Categorization of Crowdsourcing Platform Types

One possibility to categorize crowdsourcing platforms is based on the type of tasks - routine, complex or creative - they are focusing on. However, platforms often offer a multitude of different task types, which makes this approach unfeasible. Therefore, we introduce a more general categorization based on how much a platform abstracts the crowd from the employer. Hereby, we differentiate between *mediator crowdsourcing platforms*, *specialized crowdsourcing platforms*, and *platforms focusing on crowd provision*, which differ among each other in terms of their capabilities and main use cases. This results in individual advantages and drawbacks of the platform types in the context of their applicability for scientific use cases. Figure 2.3 illustrates the types of crowdsourcing platforms and their interactions.

Aggregator platforms can be seen as high-level type of crowdsourcing platforms. They often do not maintain an own workforce but recruit workers from different channels, like specialized platforms or crowd provider platforms. The

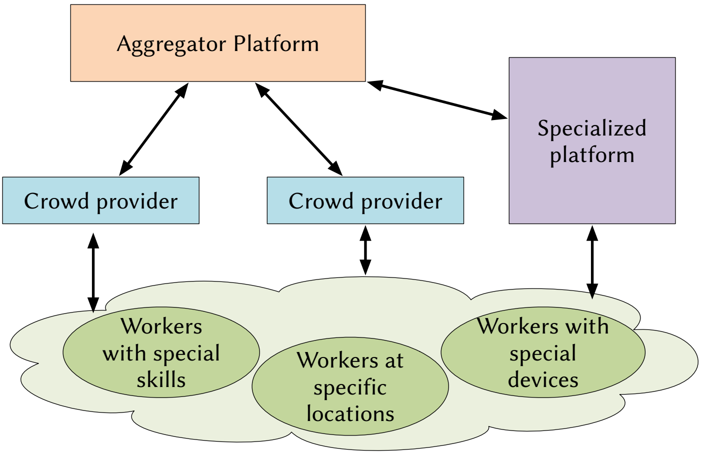

Aggregator Platform
Specialized platform
Crowd provider
Crowd provider
Workers with special skills
Workers at specific locations
Workers with special devices

Figure 2.3: Types of crowdsourcing platforms and their interactions.

main business case of these platforms is the development of crowd-based solutions for existing workflows which are not crowdsourced, yet. Therefore, the targeted employers of these platforms are usually companies trying to integrate crowdsourcing in their daily business. Besides this, aggregator platforms also offer self-service for smaller employers. Here, the aggregator platforms often focus on a specific subset of tasks for which they also offer predefined quality assurance mechanisms. The advantage of this platform type is the high abstraction of crowdsourcing related issues, like worker recruiting or quality control. Usually only the required number of submitted tasks has to be defined, the recruiting process is automated by the platform. On some platforms it is also possible to adjust the data quality via a simple slider on the platform's web interface. However, the underlying quality mechanisms are mainly optimized for simple tasks, like image tagging. The high abstraction of these platforms is also their major drawback with regard to crowdsourcing research. Due to platform internal recruiting mechanisms, the available workers might already be pre-filtered, which limits their diversity. Furthermore, the available quality assurance methods are usually not applicable for, e.g., the quality control of QoE Crowdtesting tasks. Therefore, still additional monitoring of the users is required. Aggregator platforms also add an additional business layer between the employer and the worker, which also increases the costs per task. Currently available aggregator platforms are, e.g., Crowdflower [44] or Crowdsource [45].

Similar to aggregator platforms, *specialized crowdsourcing platforms* only focus on a limited subset of tasks or on a certain type of worker. However, specialized crowdsourcing platforms have their own work force. With regard to crowdsourcing research, specialized platforms focusing on specific tasks, e.g. Microtask [46], have similar advantages and disadvantages as aggregator platforms. Due to the task specialization, self-service custom or experimental campaigns might not be possible at all. In contrast, the use of crowdsourcing platforms which focus on a specific set of workers is useful if only a limited subset of workers, e.g., from a given location or with a specific mobile device, is required [41, 47].

The most flexible type of crowdsourcing platforms are *crowd providers*, like MTurk or Microworkers. These platforms focus mainly on self-service, i.e., the employers design the tasks themselves without much assistance from platform employees, and maintain huge worker crowd. This crowd can be directly accessed through the web interface of the platform or via an API for automatic interactions. Commercial crowd providers often implement a set of filters and qualification mechanisms to select and build specialized worker groups. Due to the direct access to the crowd workers, crowd providers offer the largest flexibility in terms of task and campaign design. These platforms also accumulate a vast unfiltered number of workers from all over the world, which results in a large diversity of the potential testers. However, due to the variety of the tasks on this type of platform, the operators usually only provide a very limited set of quality assurance mechanisms and therefore advanced mechanisms must be integrated by the employer into the tasks in this case.

Table 2.2: Crowdsourcing platform categories

	Aggregator platform	Specialized crowdsouricing platform	Crowd provider
Own worker pool	No	Yes	Yes
Costs per task	Medium	High	Low
Focus on specific task set	Yes	Yes	No
Predefined quality assurance mechanisms for specific tasks	Yes	Yes	No
Unfiltered access to workers	No	No	Yes
Suitable for experimental tasks	Sometimes	Sometimes	Yes
Exemplary platform providers	Crowdflower, Crowdsource	Microtask, Taskrabbit, Streetsptr	MTurk, Microworkers

Besides commercial crowd providers, *social networks* like Facebook [48] can be used to recruit test users as well. If a task can be implemented in a joyful manner, social networks allow to easily reach a large number of test subjects for free. The task can sometimes also be integrated in a Facebook app, which additionally enables access to the users' demographic information provided in their profiles. Redesigning a task to be joyful and integrating it in a Facebook app, however, imposes a significant amount of additional work and is not always possible. Furthermore, participants recruited from a social network might be biased in terms of expectations of behaviour, if they are familiar with the creator of the task belong to the same community. Table 2.2 summarizes the main characteristics of the introduced crowdsourcing platform types.

2.2 Crowdsourcing and its Potential Impact on Future Internet Usage

Today and also within the next few years, Internet video traffic holds a significant share of the overall Internet traffic [49]. One of the key drivers for the large amount of video traffic might be the sharing features of today's social networks which connect people all over the world. These networks allow a fast spreading for information, even resulting in flash crowd effects on the content providers' infrastructure.

Crowdsourcing platforms also accumulate hundred thousands of people. Therefore, these platforms might have similar effects on the Internet traffic generation in the near future as we will show in the reminder of this section. Further, we discuss the special requirements arising for the interconnection of *human clouds* and *machine clouds*.

2.2.1 Implications of Crowdsourcing on Internet Traffic

Crowdsourcing platforms accumulate hundreds of thousands of users and process an enormous amount of tasks. MTurk reported 500.000 registered users in 2011, Microworkers about 800.000 users beginning of 2016. In December 2012, the crowdsourcing Platform Crowdflower stated to have completed about 770.000.000 individual micro tasks since 2007.

The pure amount of their users and the current growth rate of Crowdsoucing platforms make them likely to influence the generation of Internet traffic in the future. One example is Google's reCAPTCHA service [50, 51], which helps digitalizing books, identifying house numbers on Google Maps, and generating training data for machine learning systems by using Internet users to transcribe images. Even if the individual picture send to the users is just about 3KB, a daily traffic of over 90 GB is generated, as there were more than 30 million pictures processed per day already back in 2012. With today's applications of crowdsourcing, like video summarization, video tagging [52], or large scale studies for the evaluation of video quality [53], the amount of traffic generated by crowdsourcing platforms is significantly higher.

Crowdsourcing might not only generate a large amount of additional traffic, but also leads to traffic distributions that are more difficult to handle. Crowdsourcing platform users are often distributed all over the world [28, 54]. Figure 2.4, e.g., shows the distribution of the Microworkers-users beginning of

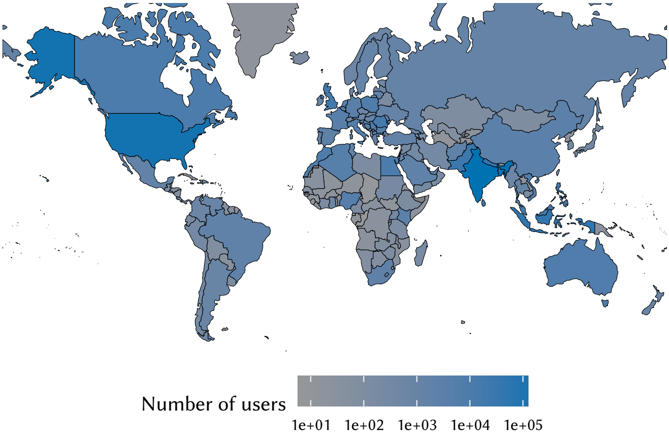


Figure 2.4: Number of Micoworkers users per country (March 2016).

2016. This world wide distribution of the users leads to a large amount of traffic between different Autonomous Systems (ASs) and Internet Service Providers (ISPs), causing similar issues to the ones in Peer-to-Peer networks before [55], e.g., high costs for the ISPs in lower levels of the ISP hierarchy.

The user distribution also leads to an activity pattern of the platforms which differs from traditional web pages. This can be observed in Figure 2.5, which depicts the activity of crowdsourcing workers on the Microworkers platform in 2016. The activity is measured by the number of tasks finished per hour of the day, normalized by the total number of tasks. The markers indicate the median worker activity, the gray shaded area the 25%, respectively 75% quantile.

Due to the different time zones of the workers, there is always activity on the server. Even though a constant utilization of the hardware resources is generally preferable, it imposes unnecessary load on the Internet as the data is always sent to the currently active worker region. To overcome this, it might be possible to exploit the regular diurnal effects for designing an allocation scheme

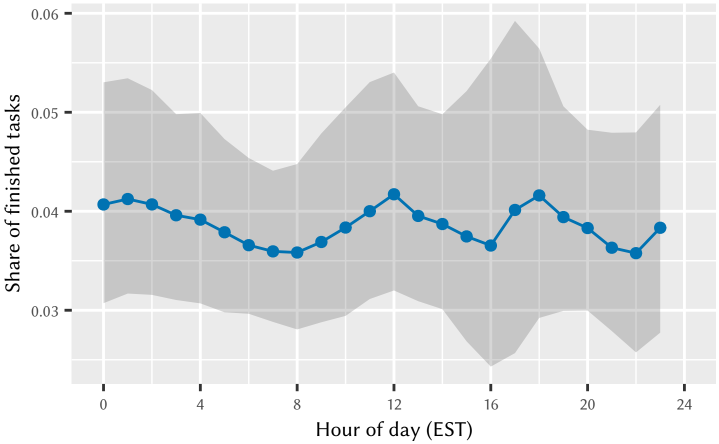


Figure 2.5: Activity of workers on Microworkers per hour of day.

of resources in a world-wide infrastructure. This would also enable the platform provider to migrate the data closer to the currently active workers, reducing their access delay and simultaneously reducing the network load. However, this requires a detailed knowledge of the usage pattern of the crowdsourcing platforms and the participating workers as presented in Section 3.1. Nevertheless, even with a well dimensioned system, still problems might occur if a large amount of new tasks is published and the human cloud workers are notified immediately. This might result in flash-crowd traffic patterns when thousands of workers connect to the platform at the same time to retrieve their task.

2.2.2 Connecting Human and Machine Clouds

The utilization of machine clouds can help to solve technical challenges like scalability and the global distribution of data. In this context, crowdsourcing can also be seen as an adaption of the cloud paradigm to human workforces.

Similar to machine clouds, crowdsourcing platforms offer an interface to access a huge easy-to-scale pool of work units which are abstracted to the user of the service. Therefore, crowdsourcing worker pools can also be considered as *human clouds*. The interconnection of human and machine clouds in turn fosters the development of completely new services. It enables the automation of tasks which requires both, high computational effort and human judgments or interactions. One example of such a task is the previously mentioned reCAPTCHA service. Here, book digitalization is realized by a combined machine-human base approach. Texts are automatically scanned and afterwards process by an OCR software. If the software is not able to determine phrases in the text, these words are submitted to a web application, which uses humans to transcribe them. This combination allows a cost effective and high quality digitalization, as the majority of the work can be accomplished by a cheap and fast machine based component and additional quality is assured by human contribution, but which in turn takes more time.

A vision for the future is an ubiquitous intercloud system that includes both machine clouds and human clouds as shown in Figure 2.6. Human clouds can be, e.g., the users of a social network or the workers of a crowdsourcing task. Currently, different initiatives and standardization organizations like the IEEE, OASIS or DMTF are on their way to standardize an intercloud architecture: Each task, application, or service that is submitted to the intercloud must include a detailed and machine-readable subtask-description [56] including technical aspects like requirements, dependencies, or tasks to perform, as well as economical and management aspects like available funds, target Quality of Experience (QoE), or maximum time until completion. With this information a special directory and mediation service in the intercloud environment will take care of finding the matching clouds that are required to compose the new cloud-based application. It also initializes communication between the different clouds and the applications and handles authentication management. Another important task that will be handled by a central directory is the trust and reputation management between cloud services. This applies especially for different labor plat-

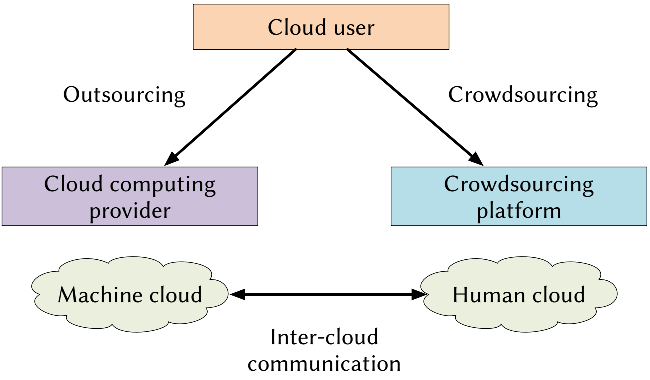


Figure 2.6: Human and machine clouds are connected by APIs and exchange traffic across the Internet.

forms which provide workforces with diverse skills. A neutral referee must keep track of the reliability and trustworthiness of the different clouds by collecting statistics and including them in the matchmaking for new service requests.

Finally, the application must be deployed in the negotiated machine and human clouds. Existing open standards and protocols for cloud communications are required to facilitate the composition of services and inter-cloud communications. An example for a system that implements important features for a future cloud architecture is NetStitcher [57], a system for stitching together unutilized bandwidth across different datacenters, using it to carry inter-datacenter bulk traffic for backup, and replication.

To develop, analyze, and compare new mechanisms that facilitate crowdsourcing and to dimension the required infrastructure, new models are required that incorporate the crowdsourcing users, the crowdsourcing tasks, and the resulting technical demands. A pure measurement driven approach is not possible in this case, due to the multiple different actors, including crowdsourcing platform providers, workers, and employers on the one side and ISPs or infrastructure providers on the other side. However, based on the knowledge about the crowd workers, their behaviour and platform dynamics, as discussed in Section 3, and exemplary task demands, e.g., derived from Section 5, basic models for dimensioning the technical infrastructure could be developed. These results could then be used to further optimize crowdsourcing platform infrastructure and the underlying network infrastructure.

2.3 Challenges for Current Crowdsourcing Systems

Besides the technical and network related challenges mentioned above, current crowdsourcing platforms also face challenges arising from the crowdsourcing approach itself. In the following we discuss some of the most common problems of current crowdsourcing systems in more detail.

Due to the large number of tasks and the high diversity of tasks on today's crowdsourcing platforms, it is often hard for workers to find tasks which match their skills and interests. Here *recommendation systems* can help to recommend appropriate tasks to workers which in turn improves the overall quality of the work. Such recommendation systems may be built by means of advanced machine learning approaches or folksonomy [58] approaches. On the other hand, specific subsets of workers could be recommended to employers during the campaign creation process. This recommendation should again be based on the skills, reliability, etc. of the workers and the requirements of the tasks.

One prerequisite for a successful matching of task and worker is the knowledge about the workers' skills and interests. This results in a need for *anonymous user profiles and specialized crowds*, which should only be accessible by the platform provider. This way, the platform provider, acts as neutral mediator between worker and employer which has the interest to operate the platform successfully for all stakeholders involved. The creation of specialized crowds out of user profiles may shorten the time it takes until a campaign is finished while at the same time the quality improves. In this direction, the question arises how to derive technical mechanisms in such a platform to automatically create and evaluate the profiles depending on the existing tasks in the platform. A further step in the direction to automate parts of the crowdsourcing process is *automated task design*. To support the mechanisms mentioned above, an automated task design may be beneficial which allows, e.g., tagging campaigns, leveraging the interaction with machine clouds and automatically finding appropriate human processing units.

Independent of the specific use case of the technical realization of a platform, the key factor of success of crowdsourcing systems is the user participation. Therefore, appropriate *incentive design* is required to encourage user participation and high quality work. This includes the relation between rewards, completion time of campaigns, and quality of work. Moreover, the targeted crowds have to be considered. In this context, the evaluation of incentives, both monetary and non-monetary, is also an important question. Micropayment-based incentives might be used for an international crowd on a web based platform, while a gamification [59] approach might work better for a mobile Crowdsensing platform in a highly developed country. Systems for real-time crowdsourcing, might again require completely different incentive approaches, like retainer approaches [60].

User profiles and recommendation systems can be used to identify skilled workers and route tasks appropriately. In conjunction with appropriate incentives, good task results are to be expected. However, still additional *quality assurance and reliability methods* are required to identify erroneous worker submission or spam submissions. While it would be desirable that the work accomplished by the human workers should be evaluated automatically, this is often not possible. If the task itself cannot be automated, an automatic result evaluation often imposes problems, too. Thus, new quality control schemes have to be deployed. Especially in the context of crowdsensing, wrong sensing values may be obtained which have to be automatically detected. While it is easy to repeat the same task several times, this creates some extra costs. Theoretical models for finding optimal but practical guidelines need to be derived.

As mentioned above, also different technical challenges exist for crowdsourcing systems related to *scalability and distribution of data.* Today, crowdsourcing platforms accumulate hundreds of thousands of users. With a future growth of these platforms similar scalability issues arise like in today's online social networks. However, in contrast to social networks, crowdsourcing platforms impose different requirements on the underlying systems as described before. Crowdsourcing platform users are distributed all over the world, resulting in a need for a global availability of the platforms data, e.g., large scale image to video data sets for tagging campaigns. Further, recent trends like real-time crowdsourcing require highly responsive systems to collect a large amount of crowds responses almost instantly.

Tightly coupled to the challenge of data distribution are also *data security* considerations. In the context of crowdsourcing, data security is not only the secure transmission of the data to the worker, but also securing the data transmitted to worker from misuse. Crowdsourcing tasks often include processing of sensitive data, e.g, hand written cheques. This data cannot simply be distributed to the anonymous crowd workers without any preprocessing. Here, methods need to be developed to identify the right amount of information so that the task can be performed correctly, but the content of the underlying document is not disclosed. To this end, images or text can be, e.g., segmented.

Furthermore, also the *privacy* of the crowdsourcing workers imposes challenges. Even if worker profiles are required to skill assessments and optimal task distribution, these profiles contain valuable private information which should not be publicly available. Especially mobile crowdsourcing or crowdsensing allows gathering even location information and mobility patterns. Here, a tradeoff needs to be found, allowing the collection of data required for optimizing task assignments and recommendation and the preservation of the user's privacy.

3 Modeling and Analysis of Crowdsourcing Platforms

In the previous chapter, we discussed several challenges of current crowdsourcing systems. A first step towards tackling these challenges is gaining a deeper understanding of the existing systems. This includes both the technical and the human aspects of the platforms.

The users of most technical systems just interact with the provided infrastructure, e.g., by watching online videos or playing interactive games. This is also true for the employers using crowdsourcing platforms to post tasks. However, the workers are a crucial component of the crowdsourcing service itself by not only using the provided platform infrastructure, but also being responsible for creating the output of the tasks posted by the employers. This results in a need to understand the human factors that influence the workers' behavior, e.g., to develop incentive mechanisms, or to optimize the task design in terms of task duration and payment.

The technical challenges of crowdsourcing platforms include the optimization of workflows and the dimensioning of the underlying infrastructure, similar to other web-based services. A prerequisite for optimizing workflows on the platforms is an understanding of the current tasks on these platforms and how users interact with the systems. The interaction patterns are furthermore a factor that has to be considered in dimensioning the systems components, besides other factors, e.g., the overall number of users.

The analysis of measurement data from existing systems helps to gain an understanding of the current status. However, to estimate further developments, evaluate different optimization strategies and new mechanisms, or dimensioning infrastructure, models of these systems are necessary. These models can either be realized as simulations or mathematical models and the level of detail depends on the research question the respective model addresses. One possible application can be an interaction model of platform users that helps the platform provider evaluating means to influence their behavior in a desired manner, e.g., to foster the growth of the platforms or stimulate the activity of the users on the platform.

In this chapter we focus on gaining a deeper understanding of existing crowdsourcing platforms and designing generalizable models of these platforms. The remainder of this chapter is organized as follows. In Section 3.1, we present an analysis of an exemplary crowdsourcing platform, Microworkers.com. This analysis includes demographic factors of the users, the tasks on the platform, and actor-centric measures like the completion times of campaigns and activity patterns of the workers. Based on the results of the analysis we present two models describing the platform growth and the activity of users on a crowdsourcing platform in Section 3.2. The findings of the chapter are summarized in Section 3.3. The content of this chapter is mainly taken from [27, 28].

3.1 Crowdsourcing Platforms Demographics and Usage

One of the main benefits of crowdsourcing is the easy access to a large group of people that is also highly diverse in terms of demographic properties. This allows researchers to conduct experiments, evaluating the impact of demographic properties, e.g., on aesthetic appeal [61], perceived quality of audio services [62], or image quality ratings [63].

To illustrate the diversity of crowdsourcing users we analyse an anonymized database snapshot from Microworkers.com, which covers the time from May 2009 when the platform was launched, to March 2016. At this time the platform had about 800,000 registered users who submitted over 261,000 campaigns and completed more than 26 million tasks.

We compare our results to findings by Ipeirotis [64–66] and Ross et al. [54, 67] who conducted several surveys with workers on MTurk to collect demographic data between 2008 and 2010. In 2008 76% of the MTurk workers are from the United States, but their numbers decreased to 47% in 2010. During the same time the share of Indian workers increased from 8% to 34%, as MTurk added the option to receive a payout in India.

Besides demographic properties, the data base snapshot also allows us to analyse the tasks performed on Microworkers and evaluate worker-, employer- and operator-centric measures. Here, we compare our results to the findings by Ipeirotis [68], who analysed MTurk from an economic point of view. Among others, he considering usage patterns of the platform and completion times of the submitted tasks. Based on data crawled from MTurk, Ipeirotis showed that 1% of the requesters post more than 50% of the dollar-weighted tasks and concludes that only a few participants make extensive use of crowdsourcing. As an additional data source, we use mturk-tracker.com to obtain recent demographic and economic data about mturk.

In contrast to existing work on demographics and usage of crowdsourcing platforms, the presented results are based on a comprehensive database snapshot from the platform operator instead of user surveys or crawled data. This enables more reliable and less biased results. We further extend the previous work by being the first to compare demographic and other platform metrics for two major crowdsourcing providers.

In the remainder of the section we first detail on the demographic background of the users, including their home countries and socio-economic properties. Thereafter, we have a closer look at tasks performed on Microworkers and the importance of big employers and very active workers. Finally, we consider different actor-centric measures of crowdsourcing platforms.

3.1.1 Demographic Background of Crowdsourcing Users

According to the definitions in Section 2.1.2, we denote to a person who has completed at least one task on Microworkers as *workers* and a person who has run at least one campaign on Microworkers as *employers*. However, we sightly extend our definition of *user*. In the following, we consider a person who has a login at Microworkers as *user*, even if this person is neither a worker nor an employer.

At first we focus on the home countries of all users. For validating a user's identity and to avoid users having multiple accounts, Microworkers sends a post card with a verification code to the worker's home address. Without a successful validation, workers are not able to receive a payment from the platform. The workers can add the full home address at any point in time prior to the first payment request, but the home country has to be submitted during the registration process. Both, home address and home country cannot be changed once they are added to the user profile. Consequently, it can be assumed that most of the information about the users' home countries are valid.

After having a general look at the home country of the users, we discuss the correlation between the prosperity of the users' home country indicated by the United Nations Development Programme's Human Development Index (HDI) [69] and their role on Microworkers. This allows us to analyse whether crowdsourcing shows the typical properties of outsourcing, i.e., employers from high wage countries use the workforce from low wage countries.

Spatial Distribution of Crowdsourcing Users, Workers, and Employers

In the following we have a closer look at the distribution of the three considered groups: *All users*, *workers*, and *employers* across all home countries and territories observed in the Microworkers dataset. Figure 3.1 shows a quantile-quantile plot of the share of users, workers, and employers versus the share of countries observed in the dataset. Note that the x-axis is in logarithmic scale. All groups show a pareto-like behavior, with about 10% of the countries accounting for about 90% of the members in each considered group. Further, we can observe a

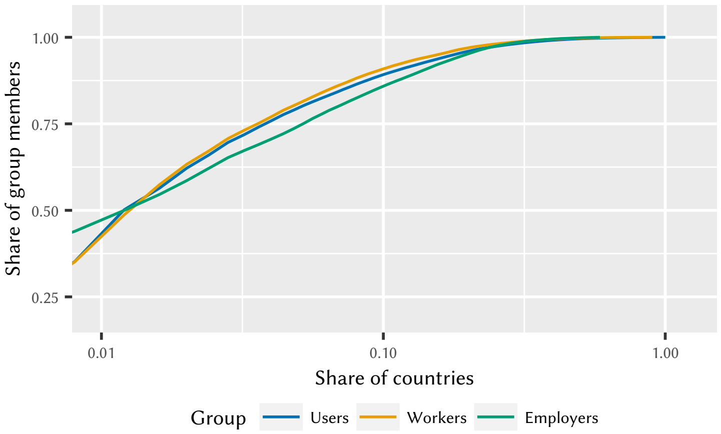


Figure 3.1: Distribution of users, workers, and employers across all observed countries on the Microworkers platform.

high a impact of a few very large countries as 1% of the countries account for about 43% of the users and workers, and almost 50% employers.

Figure 3.2 visualizes the 10 most frequent home countries for the considered groups. First, we have a look at the home countries of all users. About 76% of all users are from the labeled countries, while all remaining countries account for about 24% of the users. Most of the Microworkers users are located in Asia, with India, Bangladesh, Nepal, Indonesia, Pakistan, Sri Lanka, and the Philippines accounting for 57% of all user. The remaining users are mainly located in Europe and North-America.

Considering the most frequent home countries of the workers on Microworkers, we observe that the 10 most frequent home countries of the workers are also the 10 most frequent home countries of the users. Similar to all users, the workers are mainly located in Asia (60%) but also the United States account for a rather large amount (14%) of workers. Microworkers shows a much larger diver-

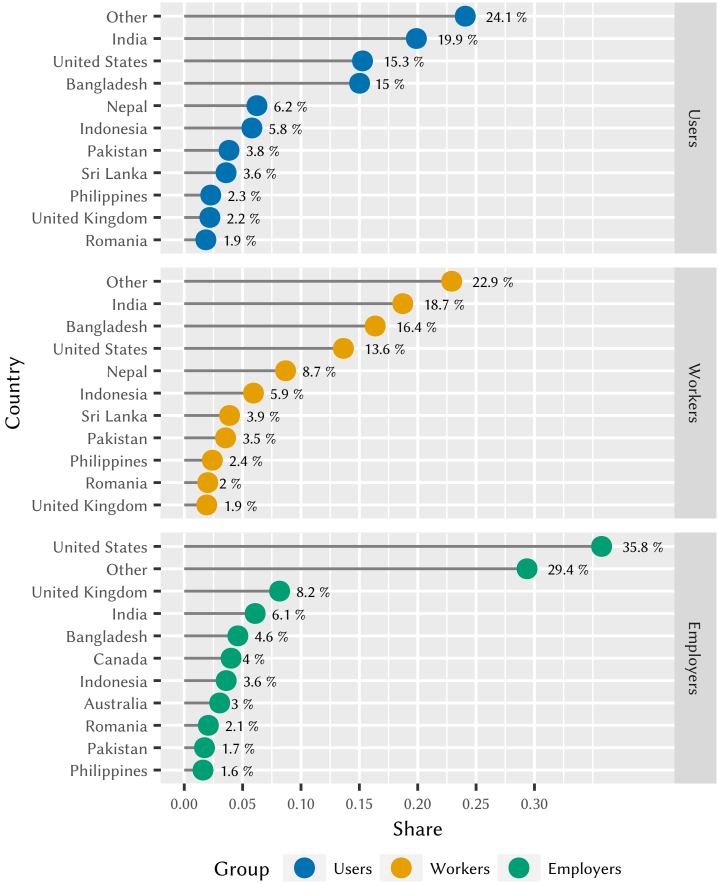


Figure 3.2: Most frequent home countries of users, workers, and employers on the Microworkers platform.

sity of the home countries of the workers than MTurk, which is heavily biases towards workers from the United States (56%) and India (36%) [54].

Considering the employers on Microworkers, we observe that the majority of the employers (35.8%) are from the United States, even the United States account only for 15% of the users. Also other high wage countries like the United Kingdom, Canada, and Australia are over-presented when comparing the share of employers to the share of users. By contrast, only for 6% of the employers are from India, even if Inda accounts for 20% of all users. The same also applies for Bangladesh. Microworkers has an international employer base, while on MTurk the employers need to be United States residents. This might result in more diverse types of tasks on Microworkers due to the larger diversity of employers. In conjunction with the international worker base, Microworkers might be a better choice for multi-lingual tasks then MTurk, since Mturk is biased towards English speaking countries.

Socio-economic Background of Crowdsourcing Users

The United Nations' Human Development Index (HDI) [69] is intended to rank countries by their level of development. A country's HDI is based on the life expectancy, literacy education and standards of living in the country. Generally, there are four types of countries: Low developed (HDI below 0.548), medium developed (HDI between 0.555 and 0.698), high developed (HDI between 0.702 and 0.798) and very high developed (HDI over 0.802). We now investigate if the home countries of the users, workers and employers are correlated with the HDI. Figure 3.3 depicts the CDF of the HDI of the home country of users, workers, and employers on Microworkers.

At first we have a look at the distribution of the workers. About 12% of them are located in low developed countries. Even if it might be possible to make a living from the money earned via crowdsourcing, there are two main factors which limit the number of workers from low developed countries. Internet access is usually only available to a few people and a certain level of English reading and comprehension skills is needed to use Microworkers. The largest share of

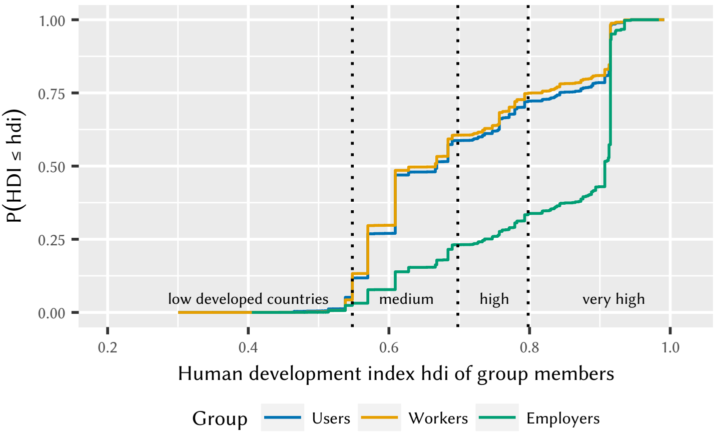


Figure 3.3: Distribution of users, workers, and employers regarding HDI.

workers (47%) is from medium developed countries. In these countries, Internet access is available to more users than in low developed countries and, compared to high and very high developed countries, the average wages are rather low. Thus, the micro tasks might be an effective way to support the costs for living in these countries. 13% of the workers are from high and 28% from very high developed countries. Here, Internet access is available to almost all people. But only a limited number of people might be willing to work on micro tasks, due to the high wages for other forms of work. However, the large number of workers from these countries shows that crowdsouring might be an accepted way to earn some extra money.

The distribution of the employers is rather different to the distribution of the workers. 77% of the employers are from high and very high developed countries and only 3% are from low developed countries. This is a typical phenomenon similar to outsourcing. The employers are located in high developed countries with high wages and thus outsource the work to low wage countries.

The HDI distribution of all users is a superposition of the CDFs for the workers and employers. However, there are about 22 times more workers on Microworkers than employers. Therefore the HDI distribution of the users is very similar to the HDI distribution of the workers.

3.1.2 Tasks Statistics in Crowdsourcing Platforms

In this section we have a close look at the tasks available on the Microworkers platform. This includes an analysis, who submits the tasks, who completes them and which tasks are typical for the platform.

3.1.3 Influence of Individual Employers and Workers

Employers have a large influence on the available tasks on crowdsourcing platforms. Therefore, we first have a look if the tasks are created by a broad range of different employers or a small group main task posters.

Figure 3.4 shows the share of reward versus the share of employers on Microworkers and on MTurk. The values for MTurk are obtained from http://www.mturk-tracker.com/#/toprequesters and consider the time between January 11, 2016 and February 10, 2016. The y-axis shows the share of all money spent, the x-axis shows the share of employers. Note that the x-axis is in logarithmic scale. We clearly see that there is a small number of employers who accounts for most of the work on Microworkers and MTurk. On both platforms 10% of the employers spend about 90% of the money. This was also previously observed for MTurk by [68] in 2010. However, the influence of large employers on Microworkers just grew over the past few years. Here, 10% of the employers only accounted for 70% of the money spent in 2010 [28]. This change might be an indicator for a changing employer type on Microworkers. While in the past most employers on Microworkers were likely to be self-employed or smaller companies using the platform for their own tasks, the current employers are likely to be task mediators building services on top of the platform.

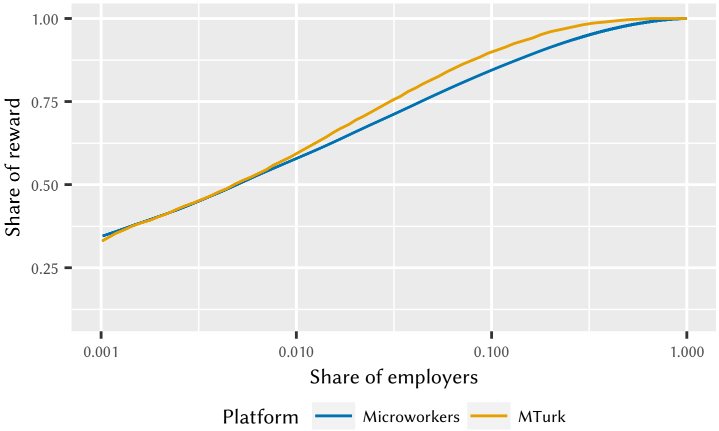


Figure 3.4: Share of reward versus share of employers.

Another factor influencing which task can be completed on a Crowdsouring platform, is the diversity of available work force. Again the platform might rely on a small number of very active workers or on a diverse crowd. The activity of the workers is measured by their number of completed tasks. Figure 3.5 shows the percentage of completed tasks versus the percentage of workers. Again the x-axis is in logarithmic scale. Similar to the activity of the employers, a small number of workers (10%) account for the majority (90%) of the completed tasks.

Task Characteristics

Next, we have a look which types of tasks are offered on the Microworkers platform. In MTurk, tasks can be classified by the given keywords or by a manual classification based on the task description. In Microworkers, a category has to be assigned to every new campaign upon creation. Each campaign is rechecked

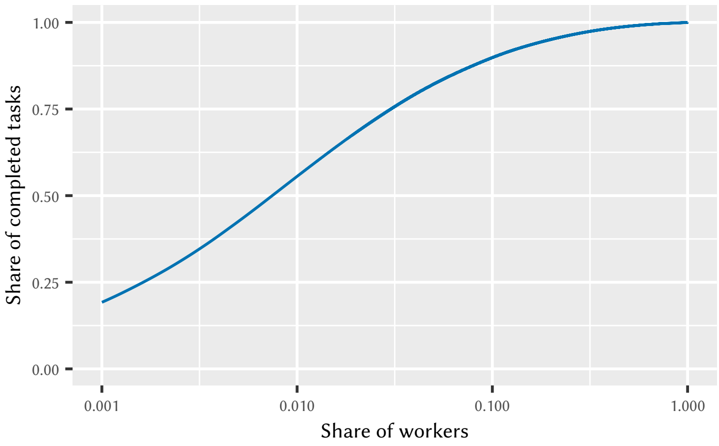


Figure 3.5: Share of completed tasks versus share of workers.

by an employee of Microworkers so that we can assume that the tasks are labeled with the correct category. Table 3.1 lists the available categories at the time of the analysis in 2016.

Most task on MTurk are related to data collection, data annotation, and gathering of subjective ratings [68]. In contrast, Microworkers was mainly focusing on search engine optimization (SEO) tasks in 2010 [28]. In 2016, the available categories of Microworkers show that the range of supported tasks has significantly increased and more complex tasks are now available on the platform. However, still a large share of tasks is SEO-related.

Further, we can observe that the reward is highly dependent on the type of task. 1.14% of all tasks belong to the category *Sign up* accounting for 0.96% of the reward and having are rather low average payment of 0.15 USD. Tasks for *Blog/Website Owners* account for about the same amount of tasks (1.43%), but for a significant higher share of the total reward (6.98%). Further, these tasks are quite well paid with 0.93 USD on average.

Category	Share of all tasks	Share of total reward	Average reward per task
Qualification	0.03%	0.01%	$0.04
Sign up	1.14%	0.96%	$0.16
Content Moderation	0.02%	0.03%	$0.25
Transcription	0.00%	0.00%	$0.24
Data Mining/Extraction	0.07%	0.06%	$0.16
Search, Click, and Engage	37.96%	32.76%	$0.16
Bookmark a page	7.05%	4.99%	$0.13
Google (+1)	3.38%	3.20%	$0.18
Youtube/Vimeo/Dailymotion/Vevo	7.73%	5.20%	$0.13
Facebook	6.87%	6.70%	$0.18
Twitter	2.76%	2.99%	$0.20
Instagram	0.05%	0.05%	$0.17
Snapchat	0.00%	0.00%	$0.19
Promotion	6.90%	5.44%	$0.15
Yahoo Answers/Answebag/Quora/Wikians	0.85%	1.11%	$0.25
Forums	1.95%	2.75%	$0.27
Download, Install	13.18%	11.37%	$0.16
Comment on Other Blogs	0.46%	0.54%	$0.22
Write an honest review (Service, Product)	0.34%	0.99%	$0.56
Write an Article	0.08%	0.49%	$1.12
Mobile Applications (iPhone & Android)	0.68%	2.24%	$0.62
Blog/Website Owners	1.43%	6.98%	$0.93
Leads	0.22%	1.54%	$1.36
Surveys	0.34%	1.18%	$0.67
Testing	0.33%	0.54%	$0.31
Other	6.18%	7.88%	$0.24

Table 3.1: Campaigns categories on Microworkers.

The payment for work usually dependents of its duration and complexity. However, a numerical analysis showed that on Microworkers the payment and the duration of the tasks is uncorrelated, with a correlation coefficient of 0.007. Tasks on Microworkers are very short but differ in their complexity or in their prerequisites, e.g., the workers need to be a blog owner or willing to submit some private data. Thus, the lowest paid tasks are simple ones like clicking an advertisement or bookmarking a webpage. Creative tasks like writing an article are paid significantly better, as well as tasks where the worker has to fulfill specific prerequisites, e.g. owning a blog. Similarly the reward for *Lead* tasks is also rather high, even if not certain qualification is required. However, the worker need to be willing to sell private data.

3.1.4 Actor-Centric Analyses

In this section we have a look at the Microworkers platform from a worker-, employer-, and platform-centric perspective, by considering relevant analyses for each of the actors. This includes, the correlation between income and the number of performed tasks, the completion time of campaigns, and the activity of the users on the platform.

Worker-centric perspective

From a worker's point of view it is important to maximize the income. This can be achieved be either focus on high payed tasks that require higher skills or focus on a large number of simple task. Figure 3.6 shows the income of the workers on Microworkers in dependency of the number of tasks they submitted. Both, the number of finished tasks and the earned reward are normalized to 1. The color of the facets indicates the amount of workers having the same number of tasks finished and earned the same reward.

We observe a clear dependency of the earned reward and the number of completed tasks, the correlation coefficient between earned reward and the number of completed tasks is 0.96. Further, the findings from Figure 3.5 are confirmed,

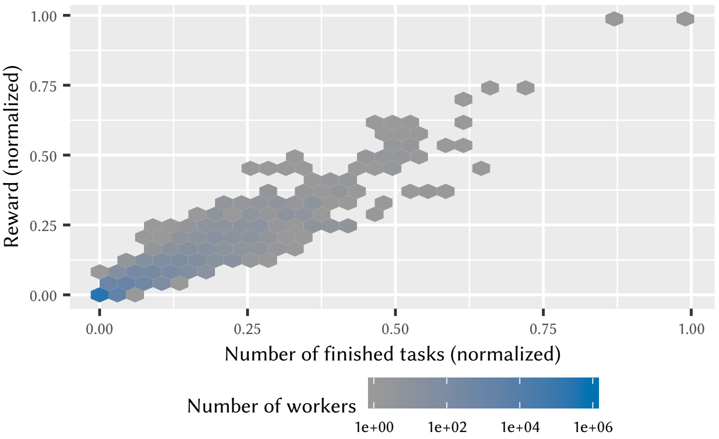


Figure 3.6: Reward in dependency of the number of completed tasks.

i.e., a relatively small amount of workers completes most of the tasks. Due to the strong correlation between the number of completed tasks and the earned reward, these power workers also earned by far the highest total reward. Having a look at workers who earned a reward of 0.25, we see that the number of finished tasks rages from 0.1 to 0.4, i.e., the number of finished tasks differs by 75% for the same total reward. This can be explained by the different payments per task category discussed in the previous section. Consequently, both strategies for optimizing a workers income are present at Microworkers. Some workers focus on a rather small number of high paid tasks, while most of the workers try to maximize their income by completing as many tasks as possible.

Employer-centric perspective

From an employer's point of view it is important to get the submitted work done correctly and fast. Quality control is a major challenge in crowdsourcing in general, thus we will have a closer look at in Chapter 4 separately. Here, we focus on the speed at which campaigns are processed. In order to assess the speed of the worker, the employer can measure, (1) when the first task was submitted and (2) when the last task was submitted, i.e. when the campaign was finished.

On the Microworkers platform, a campaign is paused as soon as all open tasks have been processed by workers. Thereafter, the employer can rate the individual tasks to be either correct or incorrect. This can take up to a few days. All tasks rated to be incorrect are then again submitted to the crowd. After all tasks are completed correctly, the employer still has the possibility to add new positions. This can again be minutes, hours, or days after the first set of tasks was completed. These functionalities make it impossible to exactly identify *the last task* of a campaign using our available dataset. Thus, we focus on the time until the first task of a campaign is submitted in the following.

Figure 3.7 shows the distribution of the time until the first task is submitted. Both x and y-axis are in logarithmic scale. We only consider campaigns, where at least one task was submitted. Campaigns with no task submissions within the first 24 hours are very rare ($> 0.07\%$). 90% of the campaigns have at least one task submission within the first 6 days after creation. The longest time observed between the starting of a campaign and the first task submission is about 5 weeks. The distribution shows that the workers respond to most of the submitted campaigns very quickly and only a few campaigns are adopted very lately. Here possible reasons are, unclear task descriptions or very high skill requirements.

We now have a closer look on when the workers submit the tasks. Figure 3.8 depicts the distribution of the submission times of the tasks. We use the timezone of the servers (Eastern Standard Time EST) as these timestamps are also used on the Microworkers webpage. Each area of the curve accounts for the continent of the submitting worker.

At first we have a look at the overall shape of the curve. The share of finished tasks per hour has two minima at 8AM and at 10PM. Most of the tasks are com-

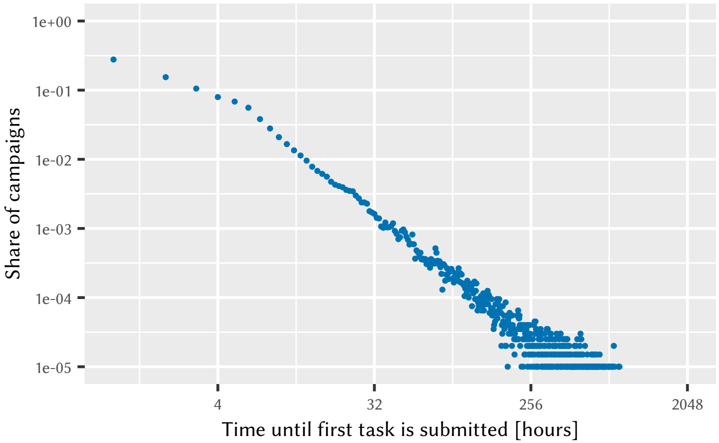


Figure 3.7: Time until the first task is submitted to a campaign.

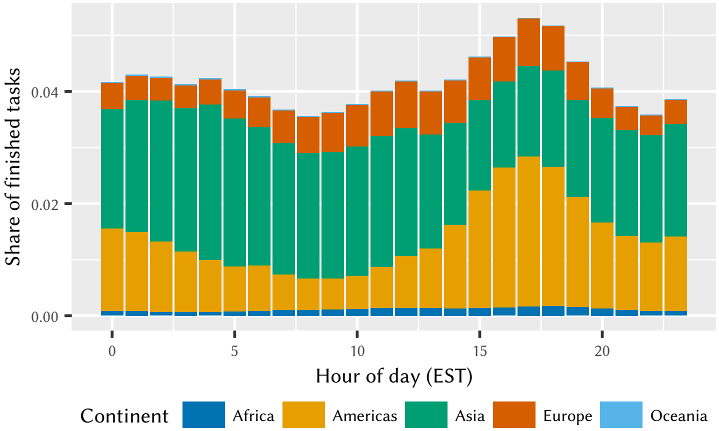


Figure 3.8: Submission time of finished tasks.

pleted between 3PM and 8PM. This behavior can be explained by considering the contribution of the workers from the different continents and consequently different time zones.

Asian workers are mainly active between 3AM to 6AM EST. Considering the most frequent home countries of these workers - India, Nepal, and Bangladesh - and the resulting time off-set of 10.5 to 11 to EST, this corresponds to 2PM to 5PM local time in e.g. Dhaka, Bangladesh. The minimum activity of Asian workers can be observed at about 4PM, which would be 3AM in Dhaka. However, during this time the maximum activity of the American workers - mainly US - is reached. Despite the different countries of origin, Asian workers and workers from America reach the peak and minimum activities at about the same time, when considering local time. Still, the activity of Asian workers remains more constant thought a day compared to American workers. For Asian workers, the minimum and maximum activity differs by about 44%, while it differs by 80% for American workers.

For European workers we can see a different behaviour, as the activity has two maxima at 12PM and 5PM that corresponds to 6PM and 11PM CET. Consequently, these workers tend to work late in the evening or at night instead during the afternoon. However, European workers play only a minor role when considering the share of completed tasks on the Microworkers platform. The share of tasks finished by workers from Oceania and Africa can be neglected.

The results of this analysis have two main consequences for an employer. First, a campaign should be submitted at about 3PM to guarantee a fast completion of the tasks. Second, the submission time of the campaign influences significantly the demographic properties of the workers working on the campaign leading to different task results.

Platform-centric perspective

From a platform provider perspective, the main goal of the operator is assuring a stable operation of the platform. This requires an appropriate dimensioning of the underlying technical infrastructure that can be achieved if the

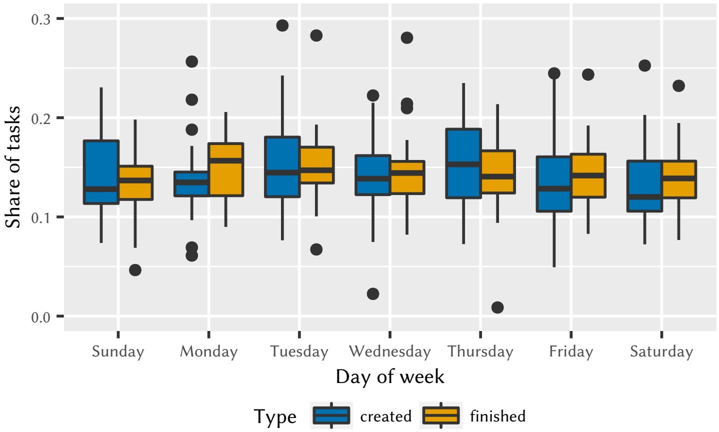


Figure 3.9: Share of tasks created and finished per weekday.

resource demands are known. Consequently, the operator is interested when the campaigns are submitted by the employer and when the finished tasks are submitted by the worker.

We already know that the number of finished tasks varies during the day due to the different time zones of the workers. This can be used for dynamic resource allocations on a short time scale or for scheduling daily management tasks to periods with low resource utilization. Still we do not know if and how the activity of the users changes during the week.

Figure 3.9 shows a box plot of the percentage of newly created tasks by employers per weekday and finished tasks by the workers. The box indicates the 25%, 50%, and 75% quantile, the whiskers extend to the largest, respectively lowest value within the 1.5 inter-quantile range. All other values are marked as outliers.

First, we have a closer look at the employers. The share of created tasks remains almost constant on Microworkers during the week, on MTurk it changes during the week [68]. This could be explained by the type of employer using MTurk and Microworkers. MTurk may mainly be used by companies which do not submit tasks at the weekend, by contrast Microworkers might be used by a mixture of companies with fixed working hours and self-employed who do not stick to fixed office hours. The workers on Microworkers show a similar behavior then the employers, they also complete the tasks almost constantly during the week. On the one hand this might be caused by workers, who work just for fun in their free time. But as the main workforce is located in low wage countries, it is more likely that these workers depend on the money and are willing to work also during the weekends.

The constant submission rate of tasks during the week causes a constant load on the server. On one side, this constant demand makes it easier to dimension the server resources, since it is not required to adapt the resources during the week, e.g., shut down servers during the weekend. On the other side, the same amount of full-time staff required each day for answering service requests and disputes related to task and campaign submissions.

3.2 Modeling Crowdsourcing Platforms

Crowdsourcing platforms including the involved actors form complex systems, so that only certain aspects can be considered in a model. However, models help the actors to evaluate different strategies to optimize their workflows and strategies to maximize their profit. Current work often focuses on the employers' point of view. Faradani et al. [70] used a crawled dataset from MTurk to model the arrival process of workers. This model was then used to design optimal pricing strategies for employers. Wang et al. [71] also used crawled data from MTurk to evaluate the completion times of campaigns. They were able to identify different factors influencing the completion times, which again could be optimized by employers. Bernstein et al. [60] optimized the costs and completion time of tasks using a model of their proposed crowdsourcing retainer approach and showed the feasibility of the approach in a proof-of-concept implementation.

In contrast to existing work, we focus on crowdsourcing platform aspects that are relevant from the provider's point of view: The growth of the user base and the activity of the platform users. To evaluate these two aspects, we first analyse the growth of the Microworkers population and show that well-known growth models applied in a range of fields including biology and sociology are capable of modeling the development of the platform's population. In a second step, we investigate the platform dynamics by developing a fluid model which is an extension of the SIR (Susceptible-Infected-Recovered) model of epidemics.

3.2.1 Population Growth

The scope of this section is to model the population growth of a Crowdsourcing platform. We use the Microworkers population data from 2009 to 2010 in order to evaluate well-known growth models with regard to their capability of describing the observed population changes. The population date from 2010 until 2016 is then used to evaluate the precision of the growth predictions of the different models. In particular, four existing population growth models are revisited in order to describe the number of users over time who registered at the Microworkers platform, which we detail on in the following.

As growth models, we consider (1) unbounded models for exponential, hyperbolic, and square growth, as well as (2) the bounded logistic growth model which leads to a maximum number of users in the system. The resulting population curves according to these models as well as the measured number of registered users over time are depicted in Figure 3.10. The parameters of the different models are obtained by minimizing the least-square errors between the measurement and the model data using the Levenberg-Marquardt algorithm [72] provided by Matlab. The goodness-of-fit in terms of coefficient of determination R^2 as well as the parameters of the different models are given in Table 3.2. The time t is measured in days and normalized to the launch date of the Microworkers platform.

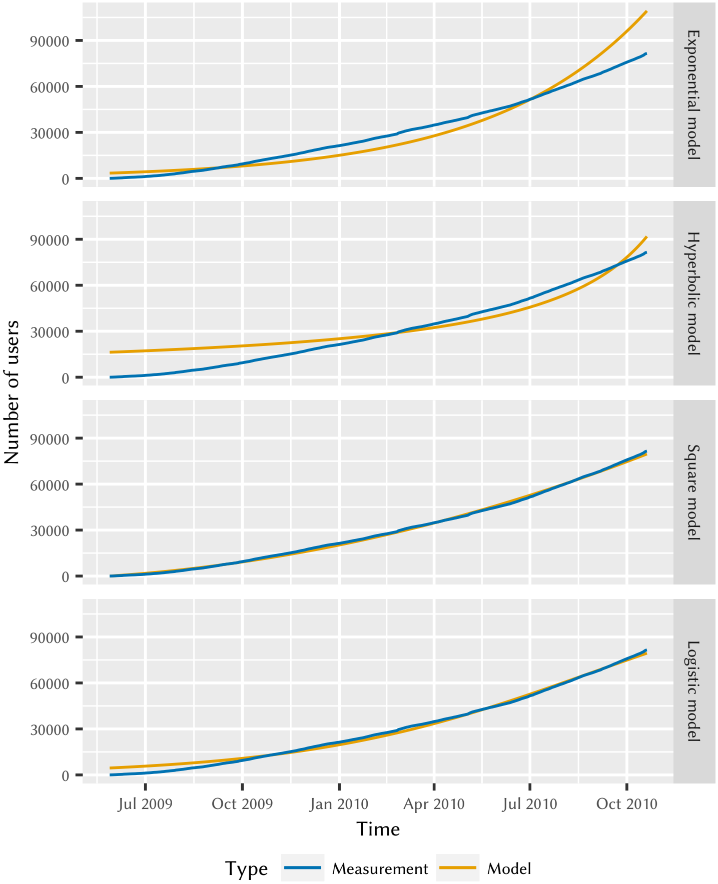


Figure 3.10: Growth models for the number of users on Microworkers.

Table 3.2: Goodness-of-fit for Microworkers growth models.

Model	Function	Parameter	Gof R^2
Square	$N_{\text{squ}}(t) = at^2 + bt$	a=0.216173, b=45.87	0.9988
Logistic	$N_{\log}(t) = \frac{N_0 \cdot K}{N_0+(K-N_0)e^{-r_0 t}}$	N_0=4507.4, K=133162, r_0=0.00734	0.9959
Exponential	$N_{\exp}(t) = N_0 e^{rt}$	N_0=3434.76, r =0.00678	0.9572
Hyperbolic	$N_{\text{hyp}}(t) = \frac{K}{t_c - t}$	K=10099646, t_c=619.95	0.9302

Unbounded Growth Model

The *exponential growth model* is associated with Thomas Robert Malthus (1766-1834) and is used to describe the growth of bacteria. The growth of the population $\frac{d}{dt}N_{\exp}(t)$ depends on the actual number $N_{\exp}(t)$ of bacteria or users in the case of crowdsourcing which are already existing at time t, i.e. $\frac{d}{dt}N_{\exp}(t) \sim N_{\exp}(t)$. It is defined by

$$N_{\exp}(t) = N_0 e^{rt} \,. \tag{3.1}$$

The parameter N_0 describes the initial population at time $t = 0$. The model parameter r is called *Malthusian parameter* or *population growth rate* which determines the outcome of the model. For $r = 0$, the population does not change. For $r < 0$, the population exponentially declines, while for $r > 0$ the population exponentially increases However, the exponential model does not apply to the Microworkers population, as we can see from Figure 3.10 and Table 3.2. In particular, the exponential growth model overestimates the number of Microworkers users after July 2010.

The *hyperbolic growth model* has a singularity in finite time which grows to infinity at a finite time t_c. This model was suggested to describe the world population until the early 1970s. It is defined by

$$N_{\mathrm{hyp}}(t) = \frac{K}{t_c - t}\,, \tag{3.2}$$

where K is a scale factor. The absolute population growth rate in the moment t is proportional to the square of the number of users $N_{\mathrm{hyp}}(t)$ at time t, i.e. $\frac{d}{dt}N_{\mathrm{hyp}}(t) \sim N_{\mathrm{exp}}(t)^2$. However, the hyperbolic model also does not apply to the Microworkers population, especially as the singularity would already appear in February 2011, but the number of Microworkers users is still far away from infinity.

The *square growth model* is not often observed in nature. The interpretation of this model is that the growth rate $\frac{d}{dt}N_{\mathrm{squ}}(t)$ of the population increases linearly over time, i.e. $\frac{d}{dt}N_{\mathrm{squ}}(t) \sim t$. The model is described by

$$N_{\mathrm{squ}}(t) = at^2 + bt \tag{3.3}$$

with the square growth factor a and the linear growth factor b. This model fits very well with the growth of the Microworkers population and no difference between the measurements and the square growth model can be seen in Figure 3.10. The coefficient of determination R^2 is very close to 1 indicating a very good match between model and measurements.

From a mathematical point of view, square growth seems to be valid for Microworkers, however, the model is unbounded which means that the number of users is not limited. Possible explanations for unbounded growth are the exponential growth of world population.

Bound Growth Model

The logistic function is applied in various fields like biology, sociology or economics, and especially in demographics for describing population growth. The logistic growth model was developed by Pierre Verhulst (1804-1849) who suggested that the rate of population increase may be limited, i.e., it may depend on population density,

$$r(t) = r_0 \left(1 - \frac{N(t)}{K}\right) . \tag{3.4}$$

In the beginning, growth is approximately exponential which slows down when saturation begins, while finally growth stops. The parameter r_0 is referred to as the *intrinsic growth rate* and is the maximum possible rate of population growth. The parameter K is the maximum number of users in the system. The dynamics of the population is described by the differential equation

$$\frac{d}{dt} N_{\log}(t) = rN(t) = r_0 N(t) \left(1 - \frac{N(t)}{K}\right) , \tag{3.5}$$

which has the solution

$$N_{\log}(t) = \frac{N_0 \cdot K}{N_0 + (K - N_0)e^{-r_0 t}} \tag{3.6}$$

with the initial population size N_0. The logistic curve converges towards $\lim_{t\to\infty} N_{\log}(t) = K$. In the case of Microworkers, it is $N_0 < K$. Thus, the population increases until it reaches the maximum capacity $K = 133162$. Similar to the square model, the logistic model fits the observed values very well.

Platform Size Prediction

The presented models were based on observations of the platform size between 2009 and 2010. More recent data allows us to evaluate the accuracy of the derived

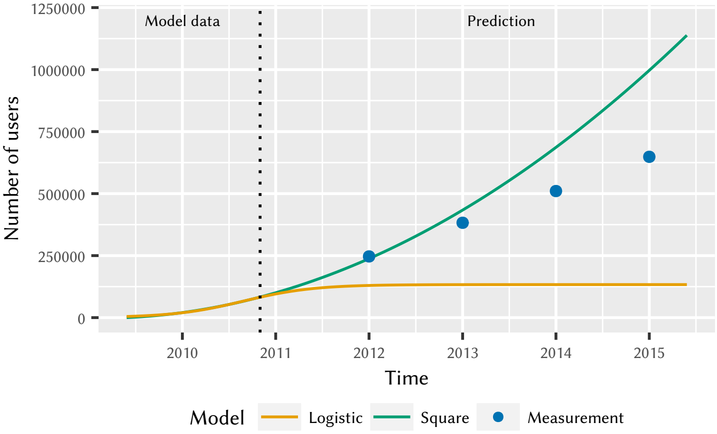


Figure 3.11: Future population growth of Microworkers.

models. The results are illustrated in Figure 3.11. We observe that the square growth model predicts the actual platform growth quite well until 2013. After this period, the square model overestimates the number of users and predicts about one million Microworkers users beginning of 2015. The logistic model significantly underestimates the number of users already in 2012. According to this model the platform population would have never exceeded $K = 133162$.

The deviation of the growth models and the actual platform size have different possible reasons. On the one side, factors outside the platform influence the number of newly registered users. Positive blog posts or news paper articles about the platform attract a large group of new users in a very small amount of time. In contrast, better working conditions on other newly emerging competitors decrease the growth rate of existing platforms. On the other side, similar factors can be found on the platform itself. New features lead to a faster growth, while changes in the terms of service might have a negative impact on the attractiveness of platform.

However, these complex and indeterministic events can only hardly be integrated into a mathematical growth model as presented here. Still the results show that a short-term to mid-term extrapolation of the number of users is still possible using the logistic model or the squared model.

3.2.2 Platform Dynamics

The scope of this section is the investigation of the platform dynamics of users becoming active and inactive, respectively. In particular, we evaluate whether the crowdsourcing platform gets successful and after which time. This is achieved by considering the steady state of the system. In addition, we analyse how the dynamics can be influenced by means of advertisement campaigns and what is the impact of this advertisement.

The platform dynamics in terms of number of active and inactive users can be described by using a deterministic fluid model. We assume that there is a fixed maximum number K of users in the system. This means that we consider a fixed population consisting of the number N of non-Microworkers users, which haven't signed up so far, the number A of active users, and the number I of inactive users.

$$K = N + A + I \, . \tag{3.7}$$

The active users have signed up to Microworkers and are actively using the platform, either as employer or worker, while the inactive users are also registered but did not use their accounts for several months. Figure 3.12 illustrates the population dynamics of the crowdsourcing platform.

Non-Microworkers users sign up and register at the crowdsourcing platform with a rate of λA. This is analogous to the basic SIR model by W. O. Kermack and A. G. McKendrick (1927), where susceptible, infected, and recovered individuals are considered in order to describe the transmission of disease through individuals. Each individual in the population has an equal probability of contracting the disease with a rate of λ. Then, all infected individuals are able to transmit the disease to susceptible individuals with rate λA, with A being the number of in-

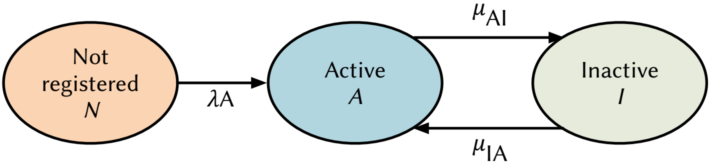


Figure 3.12: Extension of the SIR model for platform dynamics.

fected people. The application of the SIR model to the crowdsourcing platform can be interpreted as follows. The active users A of the crowdsourcing platform infect non-Microworkers users N with crowdsourcing at rate λA. Hence, the rate of new infections, i.e. users subscribing to Microworkers, is λAN. The dynamics of the crowdsourcing population can be derived with the following differential equations. It has to be noted that all variables are time-dependent, however, for the sake of readability we use a shorter notion, e.g. A instead of $A(t)$ or λ instead of $\lambda(t)$.

$$\frac{dN}{dt} = -\lambda AN \tag{3.8}$$

$$\frac{dA}{dt} = -\mu_{AI}A + \mu_{IA}I + \lambda AN \tag{3.9}$$

$$\frac{dI}{dt} = -\mu_{IA}I + \mu_{AI}A \tag{3.10}$$

Active users get inactive with rate μ_{AI}, while inactive users get active again with rate μ_{IA}. There are two different reasons why an active user may get inactive. This is reflected by the *globally influenced dynamics model* and *local user decision model* which will be explained in the following.

Globally Influenced Dynamics Model

The *globally influenced dynamics (GID) model* assumes that the usage of the crowdsourcing platform is influenced by the global opinion and popularity of the platform. Thus, the more people are active, the more people will be attracted. This includes both, the non-Microworkers users as well as the inactive users. However, this will also result in the opposite effect. If there are many inactive users in the platform, which are registered, but do not actively participate in crowdsourcing, this may disappoint active users which consequently get inactive. Accordingly, active users get inactive with rate γI and inactive users get active again with rate δA, respectively.

$$\mu_{AI} = \gamma I \quad \text{and} \quad \mu_{IA} = \delta A \tag{3.11}$$

In the steady state $\lim_{t \to \infty}$, all K users are either active or inactive, i.e.

$$\lim_{t \to \infty} N(t) = 0\,. \tag{3.12}$$

In addition, the population sizes do not change anymore in the steady state, i.e. $\frac{dN}{dt} = \frac{dA}{dt} = \frac{dI}{dt} = 0$. Thus, the differential equation system in Eq. (3.10) gets a linear equation system in the steady state

$$\begin{aligned} 0 &= -\gamma IA + \delta AI\,, && (3.13) \\ K &= A + I\,, && (3.14) \end{aligned}$$

which can be solved by

$$A = 0 \vee I = 0 \vee \gamma = \delta\,. \tag{3.15}$$

A case differentiation yields to the following results for the steady state:

$$\gamma > \delta \quad \Rightarrow \quad I = K, A = 0, N = 0 \tag{3.16}$$

$$\gamma < \delta \quad \Rightarrow \quad I = 0, A = K, N = 0 \tag{3.17}$$

$$\gamma = \delta \quad \Rightarrow \quad I = I_0, A = K - I_0, N = 0 \tag{3.18}$$

Thus, if the crowdsourcing platform operator is able to ensure that more users are active than inactive, i.e. $\delta > \gamma$, then all users will actively use the platform. In practice, this can be achieved by different forms of advertisement campaigns, by a well operated platform, by solving problems between employers and worker easily, etc.

Local User Decision Model

The *local user decision (LUD) model* assumes that users individually decide to use the crowdsourcing platform. A user is dissatisfied individually, e.g., the completion time for a campaign is too long for an employer or a task completed by a worker was not accepted by the employer such that the worker did not receive any reward. In that case, the users get inactive independent of the overall platform popularity – in contrast to the GID model. The same is true for inactive users getting active again. Independent of the global opinion, inactive users will take new tasks or launch campaigns, i.e. getting active.

$$\mu_{AI} = \gamma \quad \text{and} \quad \mu_{IA} = \delta \,. \tag{3.19}$$

For the steady state, we arrive at the following equations

$$0 = -\gamma A + \delta I \,, \tag{3.20}$$

$$K = A + I \,, \tag{3.21}$$

which can be solved by

$$I = \frac{\gamma}{\gamma+\delta} K \,, \tag{3.22}$$

$$A = \frac{\delta}{\gamma+\delta} K \,. \tag{3.23}$$

The user's decisions about the usage of the platform are independent from other users' opinions according to the LUD model. As a consequence for the platform operator, he constantly has to give incentives to its users for being active to increase the rate δ.

Influencing the Platform Dynamics

So far, we have investigated the steady state how many users will finally actively participate in the platform. However, an important factor for the platform operator from an economic point of view is the time when the critical mass is reached. According to the one-third hypothesis (OTH) by Hugo O. Engelmann, a group's prominence increases as it approaches one-third of the population and diminishes when it exceeds or falls below one-third of the population. This OTH can be applied to a crowdsourcing platform, accordingly. However, since we assume different kind of popularities and opinion forming in the GID and LUD model, we decided to consider the point in time t_0 when the steady state is reached in order to compare numerical results of the GID and LUD model.

Figure 3.13 exemplary shows the evolution of populations for the LUD and GID model. The dotted lines mark t_0 for the LUD and GID model, respectively. The x-axis scaled logarithmically denotes the time normalized by the arrival rate λ, while the y-axis shows the relative number of non-, active, and inactive users. It can be seen that the curves for the non-users N are identical for the LUD and GID model, since Equation 3.8 is identical for both models. However, the number A of active users grows faster in the GID model, since the global opinion triggers the growth of A. Therefore, the steady state is reached faster for GID than for LUD. The curve for the active users for LUD is first dominated

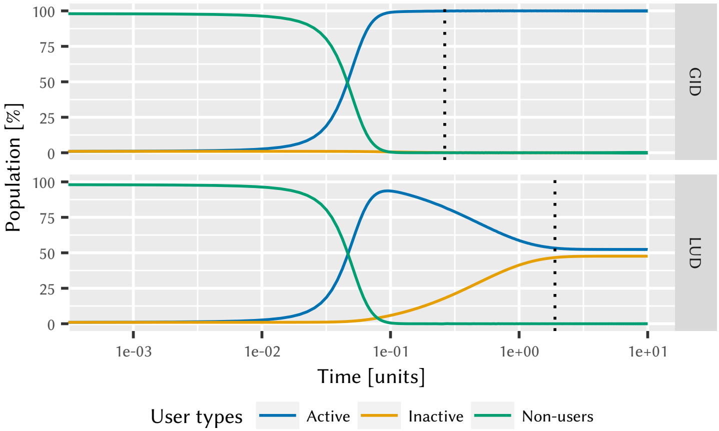


Figure 3.13: Evolution of populations for LUD and GID model.

by the arrival of non-users λ until it converges to $\frac{\delta}{\gamma+\delta}$.

A comparison of the time until the steady state is reached for both models is shown in Figure 3.14. Different platform registration rates λ are considered that are $\lambda = [0.1, 0.2, 0.3, 1]$. On the x-axis, the rate δ for getting active is varied from 1 to 10, while the rate γ is fixed with $\gamma = 1$. Hence, it is $\delta \geq \gamma$ which means that in the GID model all users will finally be active. As we can see, the steady state is always reached faster for GID for the same λ than for LUD. In addition, the step from $\lambda = 0.1$ to $\lambda = 0.2$ leads to a significant improvement for the platform operator. Thus, for reaching the critical mass fast, a platform operator should try to motivate enough people to join the platform, especially in the beginning.

Later on, the platform owner can influence the dynamics of the system by advertisement campaigns or other incentives, such that the users sign in. We consider now different registration rates $\lambda_i(t)$ reflecting different advertisement campaigns which vary in length and intensity. Nevertheless for different

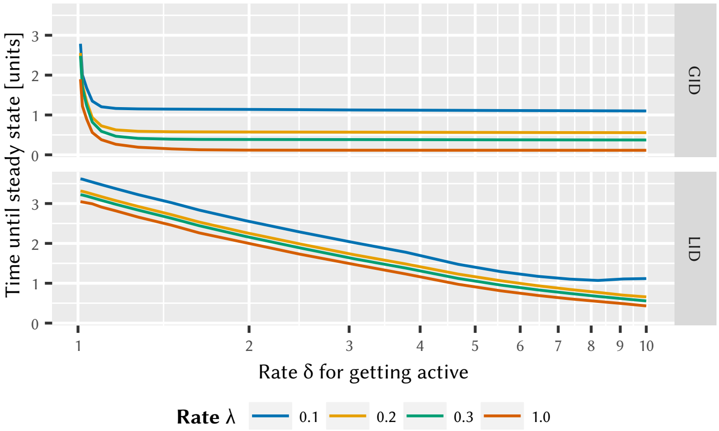


Figure 3.14: Time until steady state is reached for different platform registration rates λ.

campaigns i and j, the same number of people is motivated, i.e. $\int_0^\infty \lambda_i(t) = \int_0^\infty \lambda_j(t)$. The evolution of the system without advertisement campaign is referred to as λ_0.

$$\lambda_i(t) = \begin{cases} 0.01, & t < 0.3 \\ {}^{0.4}/_{i} & 0.3 \leq t \leq 0.3 + i \cdot 0.1 \\ 0.01, & t > 0.3 + i \cdot 0.1\,. \end{cases} \tag{3.24}$$

The influence of the different advertisement campaigns is illustrated in Figure 3.15 for $i = 1, 2, 3$ when the campaign starts at $t = 0.2$. It can be seen again that the advertisement campaign has a significant impact on the system dynamics. Furthermore, we see that the larger the peak of the advertisement campaign is, the faster the users join the platform. In summary, the platform operator has different options to influence the population dynamics. New users

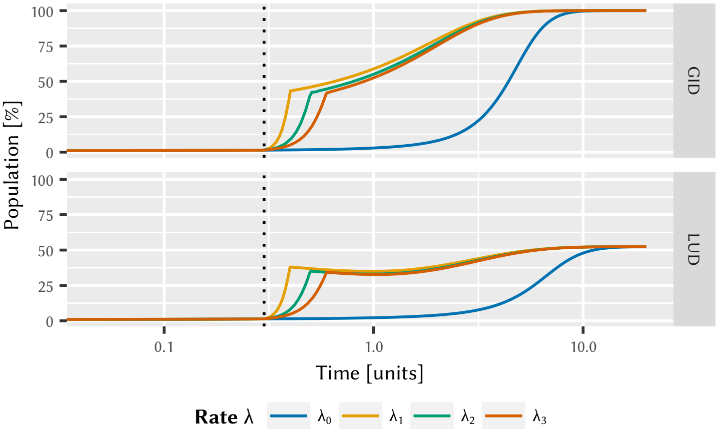


Figure 3.15: Temporal evolution of the share of active users without advertisement campaign (λ_0) and with different advertisement campaigns with λ_1, λ_2 and λ_3 starting at $t = 0.2$

should be given incentives or motivated by advertisements etc., especially in the beginning, in order to increase $\lambda(t)$. However, short, but intensive campaigns are more successful. In addition, the platform operator should foster users keeping active, i.e. increasing δ or reducing γ by appropriate means.

3.3 Lessons Learned

In this chapter, we presented an analysis of the of Microworkers crowdsourcing platform based on an anonymized database snapshot that covers the time from May 2009 to March 2016. At this time the platform had about 800,000 registered users who submitted over 261,000 campaigns and completed more than 26 million tasks. The results from the analysis were compared to existing work about MTurk [54, 64–67] and publicly available data from mturk-tracker.com.

The demographic analysis showed that 76% of the users, 77% of the workers, and 71% of the employers on Microworkers are from only 10 countries. However, the population on MTurk is even more biased. 92% of the workers are from the United States or India and only United States citizens are allowed to create an account as employer. A more detailed analysis of the Microworkers users' origin revealed that most workers (47%) are from countries with a medium Human Development Index. In contrast, most employers (77%) are from countries with a high or very high Human Development index. This indicates that crowdsourcing shows similar tendencies like outsourcing by moving work from high-wage regions to low-wage regions.

Besides the users themselves, we also had a look at the task posting and completion behavior of the users. We showed that 10% of employers spend 90% of the money on Microworkers and 10% of the workers also complete 90% of the tasks. On MTurk the influence of single employers is about the same.

Considering the three different actors on crowdsourcing platforms - workers, employers, and the platform operator - we analysed the Microworkers database snapshot according to different actor-specific metices. First, we analysed the correlation between the number of tasks completed and the income. We observed a strong correlation of 0.96 between both measures. However, workers with the same earned reward differ sometimes significantly in the number of completed tasks. This shows that there are two main strategies for maximizing the workers income. Workers can either focus on earning money through a lot of simple tasks or focus on a few tasks with higher payment. Second, we took a closer look at employer-centric metices. Our analysis showed, that workers respond to the majority of the campaigns very quickly. 97% of the campaigns have task submissions within the first 24 hours and 99% within the first 6 days. However, there are also very few campaigns that are not well accepted by the workers. Here, the longest duration between campaign start and first task submission was 5 weeks. However, details about the reasons for this long delay could not be derived from the analysed dataset. Finally, we analysed the activity patterns of the platform users as a relevant metric for employers and platform operators. It has been shown that the demographic and also the activity of the workers changes during a day. The highest activity can be observed between 3PM and 8PM EST, the minimum is reached between at 8AM and 10PM EST. This effects are caused by the different time zones and consequently different working hours of the workers. During the week, the creation rate of tasks by employers and the submission rate of finished tasks by the workers remains almost constant on Microworkers, while a weekly pattern could be observed on MTurk [68].

In the second part of the chapter, we proposed two approaches for modeling the growth of crowdsourcing platforms and the dynamics of active, inactive, and non-registered users on the platforms. We evaluated the suitability of a square, logistic, hyperbolic, and exponential growth model using the data about the Microworkers platform size between 2009 and 2010. The results indicated that a square growth model and a logistic growth model fit the measured development of the user base quite well, with an R^2 value of 0.9988 and 0.9959 respectively. More recent information about the number of users on the Microworkers platform also allowed to evaluate the accuracy of the platform growth predicted by the different models. The square model provided good results until beginning of 2013. However, non of the evaluated modes was capable of providing accurate predictions after 2013.

Finally, we presented a model for describing the dynamics of active, in-active, and non-registered users on crowdsourcing platforms. To this end we extended the SIR model of epidemics and considered two different decision models of the users, the globally influenced dynamics model (GID) and the local user decision model (LUD). We demonstrated that in the steady state of the GID model all users are either active or inactive, while in the steady state of the LUD model a fixed share of the population is active and the remainder inactive. Using the GID and the LUD model, we showed that advertisement campaigns to attract new users should be short and intensive instead of long-term with a decreased rate.

4 Optimizing Result Quality in Crowdsourcing Systems

In contrast to machine clouds, the quality of tasks results obtained from human clouds can vary significantly. Especially cheating workers, unclear instructions, or a lack of qualification can lead to low quality results. Crowdsourcing tasks are completed remotely by anonymous workers without any supervision. This anonymity can encourage workers to cheating, i.e. they try increasing their income by intentionally using malicious techniques, even if the expected gains are rather small [73]. Besides intentional cheating, issues caused by the task design can also result in low quality results [74]. However, it is difficult to identify those problems, e.g., misleading instructions, as a direct interaction between workers and employers is usually not possible.

Numerous efforts have already been made to improve the quality of the task results submitted by the workers. Most approaches try to assess the quality of an individual worker, use group- or workflow-based mechanisms to level out individual erroneous results, or optimize the task design. Guidelines for an optimal task design were, e.g., given by Kittur *et al.* [75] who conclude that cheating should take longer then completing the task properly. Eickhoff [76] suggest to discourage cheaters by an appropriate task design instead of detecting them and together with Vries [77], Eickhoff observed that depending on the type of task cheaters are encountered more or less frequently. Gadiraju et al. [78] showed that not only the task type, but also the design parameters *task length*, *monetary reward*, and *time required for task completion* influence the amount of cheaters attracted by a task.

In this chapter we support the efforts of optimizing the quality of crowdsourcing tasks results by extending existing work in two directions. First, we demonstrate an approach for assessing the quality of an individual worker, second we provide a numerical model for evaluating the costs and accuracy of two widespread quality assurance workflows. To this end, we show that an analysis of the worker's interactions with the task interface can be used to estimate the quality of the task results in Section 4.1. We use an exemplary language skill assessment task and a web-based interaction monitoring toolset to evaluate the feasibility of this approach. Section 4.2 presents an analytic model for two group-based quality assurance mechanisms. Using this model we evaluate the accuracy and also the costs for both approaches for different types of crowdsourcing tasks. Section 4.3 summarizes the findings of this chapter. Note that the content of this chapter is mainly taken from [3, 7, 20, 29].

4.1 Task Interaction Monitoring for Assessing Worker Quality

The most common way to test the trustworthiness and quality of a worker is to add gold standard data tasks [79], whereof the correct task result is already known. Gold standard data can increase the quality of the task results, since the task designer can give the worker an immediate feedback about mistakes. Further, continuously cheating workers are easy to identify and removed from the worker pool. In some cases, gold standard data can be generated automatically [80] and also the bias of workers can be taken into account [81].

Gold standard data is not applicable for tasks where there is no clear *correct* result, like a subjective rating. For such tasks, Chen et al. [82] used combinations of pair-comparisons to assess the workers consistency (intra-rater reliably) to identify erroneous submissions and developed a crowdsourcing platform for Quality of Experience (QoE) assessments. Kittur et al. [75] used crowdsourcing workers to rate the quality of Wikipedia articles. The correlation between the rating obtained from crowdsourcing and a trusted reference group could be significantly improved by adding questions testing if the worker read the article.

Most approaches assessing the workers' task quality focus only on the outcome of the task, but not the root cause of the low quality. Kazai et al. [83] introduced five worker types, *Spammer*, *Sloppy*, *Incompetent*, *Competent*, and *Diligent*, based on the observed worker behavior and a survey. They suggest that a fine granular distinction among the worker types can be used for optimizing the task design or finding appropriate workers. Gadiraju et al. [84] demonstrated that different types of malicious workers in a survey task can be identified using technical measures and propose specific counter measures for each of the identified types.

In this section, we extend existing work by showing a fine-granular monitoring approach of worker-task interactions. We demonstrate how such interactions can be analyzed using an Application Layer Monitoring (ALM) approach and demonstrate its applicability to estimate a worker's performance. Our work is closely related to the work by Rzeszotarski et al. [85]. However, we focus on a single exemplary crowdsourcing task. This enables us to develop a fine grained monitoring framework that allows us to analyse the workers' behavior in more detail and with a larger diversity of test participants. To this end, we use a simple language test as example for a crowdsourcing task, which is described in Section 4.1.1. In Section 4.1.2, we detail on a possible implementation of an ALM approach for this task. This monitoring enables us to derive fine granular temporal information, about how much time the participants spend on specific parts of our test. Section 4.1.3 discusses the results from the ALM and their interpretations in terms of worker behavior. Using these results, we show that it is possible to predict a worker's performance in Section 4.1.4. The content of this section is mainly taken from [20].

4.1.1 Language Qualification Test as Exemplary Use Case

We use an English language test to illustrate a possible implementation and the benefits of ALM in a crowdsourcing environment. Such a qualification test is not necessarily a common crowdsourcing task, however English language comprehension is a very essential qualification on crowdsourcing platforms [74] to achieve high quality results.

Test Design

The test consists of five texts with five multiple choice questions for each text, resulting in a total of 25 questions on one single web page. In order to increase the difficulty to share any solutions of the test, the order of the texts, the questions, and answers is randomized. Additionally, one text production question is added at the end of the test, where the worker is asked to state which text he likes best and why. Workers are only able to complete the task, if all questions are answered. After these mandatory questions, the worker can leave optional feedback on a separate page.

The test texts are based on slightly modified articles from the Simple English version of Wikipedia articles. The texts' topics include science, celebrities, pop culture as well as recent history, and every text contains approximately 200 words. Although the topics are rather common, we make sure that the texts contain very specific information hardly any candidate can answer due to prior knowledge.

The questions are constructed according to Day et al. [86] who give detailed advise on how to design language comprehension tests. Two questions aim at *literal comprehension*, i.e. the required information can explicitly be found within the text. One question aims at *reorganization*, i.e. extracting and combining several pieces of explicit information from the text is necessary. The two remaining questions aim at *inference*, i.e. the required information is only implicitly stated in the text and needs to be inferred. The literal comprehension questions are rather easy to solve as the answers are explicitly given in the text. In contrast, the reorganization question are more difficult, as a deeper understanding of the text is required. The inference questions are assumed to be the most difficult question type as abstract thinking is required here.

For each question, the worker is given four possible answers. In order to derive additional implicit feedback from the participants, we deploy a special answer scheme. Two of the answers can actually be found within the text, but only one of them makes sense regarding the questions and is correct. This can be used to distinguish between participants who have read the text, but did not understand the question. The other two answers sound possible regarding the question, but cannot be found within the text. These answers are intended to capture people who may very well have understood the question, but may have skipped the text to save time.

To derive the participants' score, we deploy a very simple scoring system that assigns one point per right answer and zero points for each wrong answer of the multiple choice questions. The total score is then calculated as the sum over all points. Consequently, the lowest score that can be reached amounts to zero points in total, while the highest score is 25 points. The text production question is not considered in this scoring system as it is not possible to evaluate it objectively. However, it can be used as an indicator how serious a worker is taking the test, e.g., by considering the length of the answer.

The motivation of the test is to determine whether or not a candidate has the qualification to understand English task instructions. Although our scoring system allows for a graduated assessment of this qualification, the choice of whether or not a candidate will pass the test is a binary one. Therefore, we intend to determine a suitable *Qualification Threshold* that needs to be reached in order to pass the test.

Multiple choice tests tend to foster the cheat pattern of satisficing [87], where candidates simply try to fill out the form as quickly as possible. For the subsequent considerations, we assume that these people will pick answers in a uniform distributed fashion. However, due to the fact that each candidate receives a uniform distributed random sequence of answers, we also cover people that would use a deterministic answering pattern, i.e., for instance always picking the first answer. We calculate the probability of a candidate randomly passing the test, which we also refer to as the probability of *false positive qualification*. The probability of randomly reaching k points can be modeled using a Binomial distribution, with $p = \frac{1}{4}$ as the probability of randomly selecting the correct answer and $n = 25$ questions. Thus, the probability of reaching $k = 25$ points by chance would amount to

$$P(X = k) = \binom{n}{k} p^k (1-p)^{(n-k)} = \binom{25}{25} \left(\frac{1}{4}\right)^{25} \left(1 - \frac{1}{4}\right)^{0} \approx 8.9 \cdot 10^{-16}.$$

Our desired sample size is in the order of $m \approx 10^2$ up to $m \approx 10^3$ individuals. Therefore, we decide that a probability of *false positive qualification* of approximately 10^{-3} is sufficient to make sure that (almost) none of our candidates passes the test by chance. This probability can be reached for $k > 12$ yielding in

$$P(X > k) = 1 - \sum_{i=0}^{k} \binom{25}{i} \left(\frac{1}{4}\right)^{i} \left(1 - \frac{1}{4}\right)^{(25-i)} \approx 3.3 \cdot 10^{-3}.$$

For the sake of simplicity, we henceforth normalize the maximum score to 100 percent and set the *qualification threshold* at 50 percent. Workers with test scores below 50 percent are referred to *non-qualified*, workers with test scores above 50 percent as *qualified*. Further, we are also interested in the reasons why workers fail the test, i.e., if they are trying to trick the system or if they lack the required language skills. Therefore, we do not automatically assume that every unqualified worker is also a cheater without further investigation.

Test results

For our study, we recruit 215 test candidates in February 2013 using the Microworusing.com crowdsourcing platform. The payment for the task amounts to 0.10 USD, which is comparable to that of similar studies [87].

As we are conducting a language test, we first have a closer look at the origin of the participants. The demographical information are obtained from the workers' profile pages on Microworkers. Our candidates come from 22 different nations, however, the ten most frequent countries make up about 90% of the participants. Most of the participants come from Bangladesh (41%), Nepal (10%), and Sri Lanka (10%). Besides India and Pakistan in Asia, several workers from Eastern Europe participated. About 2% of the participants are native speakers, from the United Kingdom and the USA.

Figure 4.1 depicts the workers' test results as a complementary cumulative distribution function (CCDF), with the normalized score s on the x-axis. The qualification threshold is shown as a vertical dashed line. The curve starts with a score of 0.08 and a probability of 100%, indicating that no candidate has less than 2 correct answers. On the contrary, 18% of the candidates achieve a maximum score. The curve intersects with the qualification threshold at a probability of 71%, i.e., 71% of the workers pass the test, while 29% fail.

In general it is not possible to assess a worker's result as we did here, since the correct task result is usually unknown. However, we will show that the interactions of the workers with the task interface can give first insights into the expected quality of the results.

4.1.2 Application Layer Monitoring in Crowdsourcing Tasks

For our approach, we assume that the interface of the crowdsourcing task is implemented as web application. This enables us to add monitoring components using common web development techniques and guarantees the preservation of the worker's privacy as we are only able to monitor the interactions with the tasks interface. This is similar to a regular work place, where the supervisor can observe the office space of the employees.

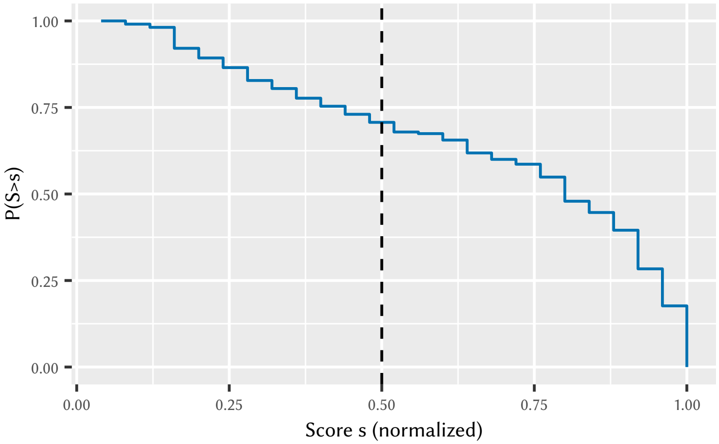


Figure 4.1: CCDF of the normalized scores s from the test participants.

General Approach

Our approach gathers information about the worker on both, client and server side. In general, server side measurements enable monitoring the accessed resources of the web application and the time of the request. In our use case, the application only consists of one single web page, therefore we use the server logs to analyze when and how often a worker accesses our page. Furthermore, the time of the submission of the form can also be derived using the server side information.

More of the information about the workers' behavior can be derived from their interaction with the application interface itself. Using JavaScript and Document Object Model (DOM) application events, each interaction with HTML elements, such as buttons, text fields, etc. can be monitored with milliseconds precision. It is also possible to monitor a limited range of interactions with the browser itself, such as changing or closing the application window as well as switching to another browser tab. This allows us to study the worker's interaction behavior on a very fine-grained interaction level including, leaving and entering the application, click behavior, mouse movement, scroll movement, manipulating interface elements and text inputs, and text selection.

Use Case Implementation

As mentioned before, ALM has to be implemented on a per task basis, i.e. the expected behavior of a worker has to be known in advance to be able to identify suspicious worker interactions. To model this expected work behavior, we consider the steps that are necessary to solve the test. Naturally, the worker starts by reading the instructions at the top of the test. In order to either get to a text or to get to the questions, the worker has to scroll. Then he reads a text or is engaged in answering questions, i.e. in finding and picking the right answer. To get to the next text passage of the test, the worker then scrolls again and so forth. We were interested in the sequence and duration of these steps as well as the details of what the worker is doing in them. In order to determine the periods in which the worker remains in a certain step, we rely on two measurements:

1) The worker's vertical scroll position used to reconstruct the worker's current field of sight (measured synchronously in intervals of 10 sec)

2) The worker's interaction with the answering elements of the survey (measured asynchronously, event-based). This particularly includes:
    - The clicks on radio buttons for multiple choice questions.
    - The selection and de-selection of the text box for the text production question.

Further, we consider how the worker interacts with the test while answering a particular question. This includes the time it takes the user to answer as well as the number of re-decisions for a specific question. Particularly high values for these variables might indicate possible difficulties users had with our task design, while particularly low values might be used as indicator for cheating behavior, such as satisficing. The measures can be derived from the aforementioned click measurements and are detailed on in the following section.

4.1.3 Evaluation of Application Layer Measurements

In this section, we review potential ALM metrics, the completion time, working phases, and the consideration time. Further, we show that these metrics can be used to analyze the work behavior we would expect.

Completion Time

Previous work has shown that the completion time of a task can be used as an indicator for the quality of task results [26, 88]. Furthermore, by defining time thresholds, low performing workers can be detected [89, 90]. In order to derive such a threshold for our task, we assume an ideal worker with the following properties:

1) The ideal worker is familiar with speed reading techniques, which allow him to read 2000 words per minute. Thus it would take about 51 sec to read all 1500 words within the test.

2) The ideal worker is able to answer any question within $t_q = 5\,\text{sec}$.

As a result, the *Plausibility Threshold*, i.e., minimum completion time would yield in $t_{pt} = t_r + 25 \cdot t_q = 176\,\text{sec} \approx 3\,\text{min}$. In our test, the completion time varies between 1.03 min and 55.95 min with a median completion time of 20.13 min. We observe that 7.6% of our participants have completion times below our plausibility threshold.

In the following we analyze the coherence between the workers' scores and the task completion time, which is visualized as a scatter plot in Figure 4.2. The x-axis describes the completion time in minutes, while the y-axis denotes the score normalized to 100%. Each data point represents the performance of a single

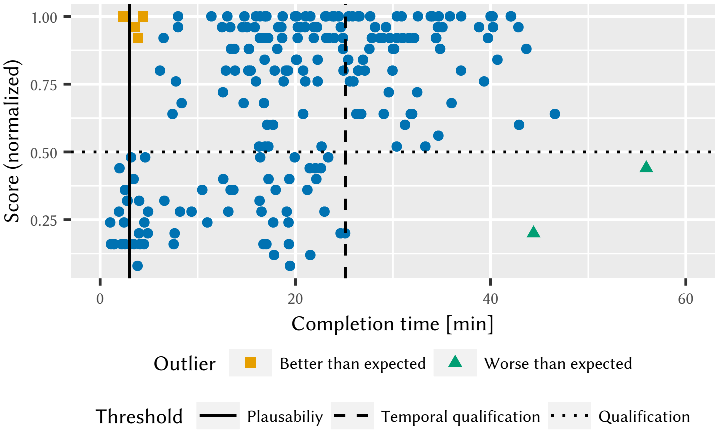


Figure 4.2: Completion time and test score per worker.

worker. The qualification threshold is included as a horizontal dotted line, the plausibility threshold as a vertical line.

We observe that almost all workers with completion times below the plausibility threshold also drop below the qualification threshold. Nevertheless, our test sample includes four participants, who completed the test approximately within our expected minimal time and still achieved scores between 92% and 100%. A closer analysis of these workers shows that all of them accessed the test at least half an hour prior to their submission. It is likely that the workers copied the text to familiarize themselves with the test, respectively completed it offline in advance.

The plot also reveals that almost all workers with a completion time above 25:07 min qualified in our test. Thus, this value could be regarded as the *Temporal Qualification Threshold* for our test. Nonetheless, we observe two outliers, one at 44:24 min with a score of 20% and another at 55:57 min with a score of 44%.

Our analysis did not show any further abnormalities for these candidates, so we are not able to determine reasons for their low performance.

We conclude that for the completion time of a task, temporal thresholds are well-suited in order to give a first assessment of quality respective qualification in our case. The two temporal thresholds subdivide the plot into three horizontal segments. The plausibility threshold can predict the non-qualification of candidates in the first segment, i.e. below the threshold. The temporal qualification threshold on the other hand can be used in order to predict the qualification of workers with completion times in the third segment. However, none of the thresholds can make predictions for the second segment, which includes most of the workers. Further, the predictions based on the plausibility threshold and the temporal qualification threshold show classification errors in some cases. Moreover, the temporal qualification threshold can only be estimated if the results of the test are already evaluated and therefore, it cannot be used as an input parameter during the evaluation process itself.

In this analysis, we only considered the duration of how long workers worked on the tests. In the following we shed light on what the workers actually did within our test by investigating their low-level interactions with our application.

Working Phases

Instead of considering the time it takes the workers to complete the whole test, we now consider the time he spends on reading and answering. While completing our test task, we assume the worker to be either reading the instructions or the texts, or answering a multiple choice question or the text question. To analyze the time the workers spend in these phases we use the following estimators:

(E1) "Reading Instructions" (RI) describes the time the instructions were visible to the worker, i.e. the time the worker had the chance to read them. The estimator "Reading Text x" (Rx) works in a similar fashion for each of the texts.

(E2) "Answering the Questions for Text x" (Ax) calculates the time difference between the timestamp of the first time the worker could have possibly seen the questions about text x and timestamp of the last answer given for these questions. The estimator “Answering the Text Question” (AT) works in a similar manner, but uses the timestamp of the last de-activation of the text box as end time.

We use JavaScript to monitor the workers’ interaction with radio buttons or the text field. Here, every change triggers an event which can be captured. However, large parts of our test include reading texts. During this time, the worker does not explicitly interact with the web page. In order to analyze these reading phases, we estimate the currently visible area. This information can be retrieved using the current scroll position in conjunction with the browser window height, which can be determined using JavaScript, too. A non-responsive CSS layout which ensures a fixed size of the web page independent of the workers’ device resolution enables us to recalculate the position of the visible elements of the web page at any point in time. However, most of the time the worker has the chance to see several elements belonging to different phases. Therefore, the aforementioned estimators are constructed in a fashion which allows the different phases to overlap.

Figure 4.3 visualizes the average time the workers spend in the different phases, including the 95% confidence intervals. The nomenclature of the phases on the x-axis follows the one introduced for the estimators, whereas the numbers indicate the text or questions position. $R3$, e.g., refers to the time the workers spend on reading the third text. The y-axes denote the absolute duration in seconds, note that the ranges of the y-axes differ for both worker groups. We also included two threshold as dashed lines. The estimated minimum reading time of 6 sec for one text and the five corresponding questions, and the minimum answering time of 25 sec for one set of five questions. The minimum answering time refers here to the time it takes to complete all five multiple choice questions per text.

We can observe that none of the qualified workers drops below one of the thresholds, but spends on average between 130 and 187 sec on reading a text and even more time on answering the questions. Furthermore, the duration of the answering phases even increases with each text for the qualified workers. This

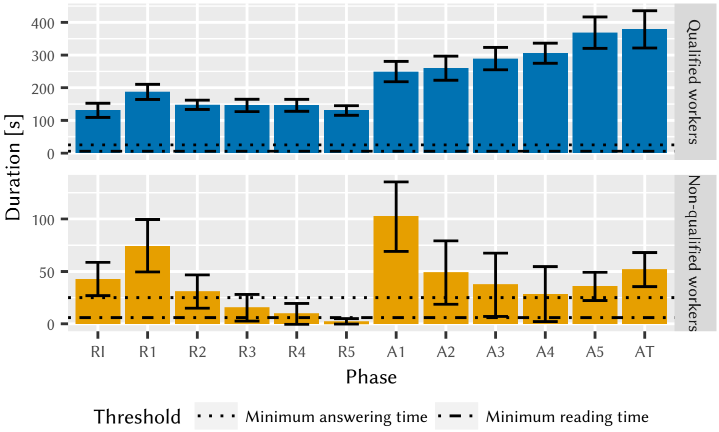


Figure 4.3: Average durations of the phases in the English language test.

indicates that they work more diligently towards the end of the test. However, the confidence intervals for all answering phases overlap, except for the first and the fifth phase indicating that this behavior needs not be typical for all of the workers. The average phase time of the text production question is even higher than the answering phase times for the questions.

In contrast, non-qualified workers tend to fall below the thresholds. Moreover, we can observe that both reading and answering phases tend to decrease with each text. Still the duration of the reading and the answering phase for Text 1 is factually higher than the corresponding duration for Text 5. This indicates that these workers' motivation might have dropped during the test.

Analyzing the working phases of the participants allows us a much stricter distinction between qualified and non-qualified workers than the analysis of the task completion time. Using the same approach of temporal thresholds but with a finer granularity, we can clearly distinguish between workers working diligent and workers only trying to complete the test as fast as possible.

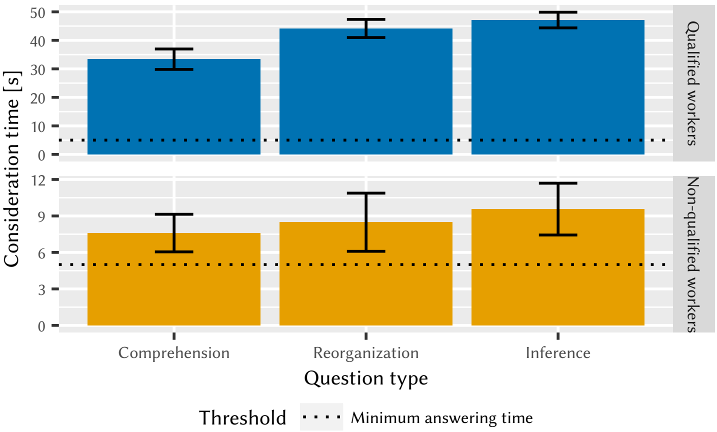


Figure 4.4: Consideration times per question type.

So far, we analyzed the overall completion time of the test and the duration of the working phases. However, the working phases consider only the time the workers spend on answering all five questions related to one text. In the next paragraph, we have a closer look at the answering process of the individual questions and analyze which information we can derive from this information.

Consideration Time

For the analysis how much time the test participants spend on the single questions, we use the variable *Consideration Time*, which is the time between the first time a worker saw a question and the time the worker changed his answer for this question for the last time.

Figure 4.4 visualizes the mean consideration times including the 95% confidence intervals for the different question types described in Section 4.1.1. The dashed line indicates the 5 sec threshold for the estimated minimum answering time. Note, that the rage of the y-axes differ. For the qualified workers, we observe the consideration times are above the expected threshold. Furthermore, there is a tendency that questions with a higher level of difficulty cause higher consideration times. This is also intuitive, as simple questions based on literal comprehension require less effort than reorganization or inference questions, therefore they can be answered more quickly. In contrast to this, the difficulty of the questions do not have a significant impact on the consideration times of the non-qualified workers. Even if the mean consideration times for the questions are also higher than the minimum answering time, it is clearly visible that the non-qualified workers spend significantly less time on answering the questions than the qualified workers.

Next, we examine the consideration time with regards to different answer types which are depicted as CDFs in Figure 4.5. The x-axis denotes the consideration time on a logarithmic scale in seconds. The curve for the answer type *plausible, in text* shows the consideration times for questions that were answered *correctly*, while the remaining curves are incorrect answers. We can observe the overall tendency that picking the correct answer requires more consideration time than picking a *wrong* one. Further, it takes more consideration time to pick an answer that is not plausible, but in the text than an answer which is plausible but not in the text. This might be caused by people who did not understand the question, but at least try to look for an answer within the text. It also becomes apparent that the answers which are plausible but in the text indeed tend to capture "lazy" people that may simply skip the text.

The results from the consideration times indicate that even on a very low level of interactions, worker behavior can be monitored and suspicious behavior can be detected. In the next section, we analyze to which extent the gathered information about the worker behavior can be used to predict a worker's test score.

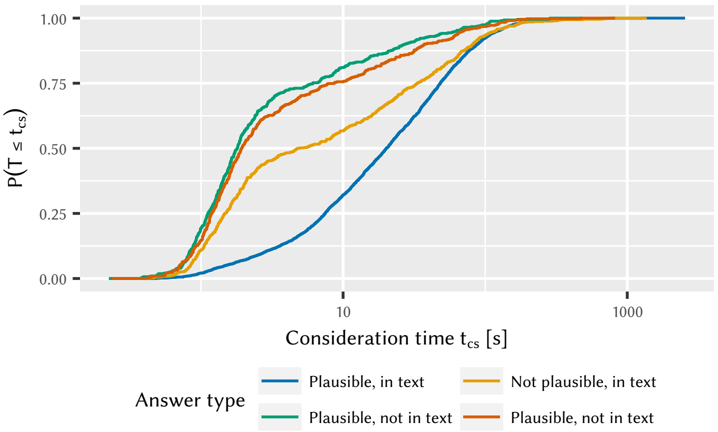


Figure 4.5: CDF of consideration times t_{cs} per answer type.

4.1.4 Predicting Task Result Quality

In contrast to most Crowdsourcing tasks, we could evaluate the task results objectively. This enabled us to define the qualification threshold and to assign each test participant to the categories *qualified* or *non-qualified*, which we used during the analysis of the interactions with the task interface. For Crowdsourcing tasks in general, the *correct* result is usually unknown. Therefore, we now want to analyse if it is possible to predict the quality of a task result, in our case the category a worker is assigned to, solely by the ALM measurements.

To achieve this, we use supervised machine learning to train a Support Vector Machine (SVM) using the features, *task completion time*, the different *phase times*, and the *mean consideration time*. An SVM uses the feature vectors of a labeled set of qualified and non-qualified workers to derive a classifier for separating both groups. This rule can then be applied to an un-labled set of workers and assign them to the appropriate group based on their feature vec-

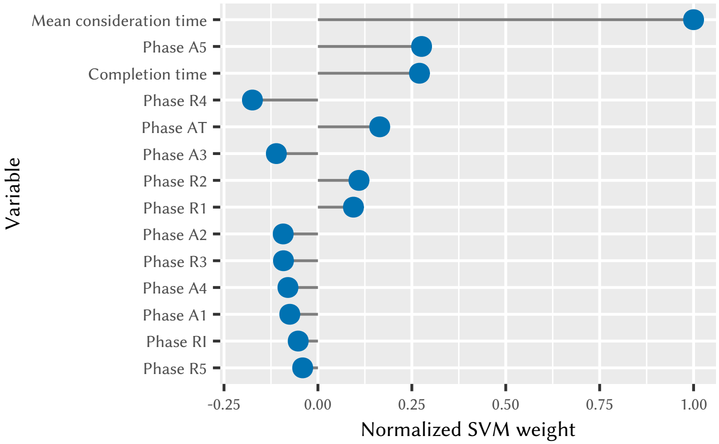


Figure 4.6: SVM weights of features for qualification prediction.

tors. To avoid over-fitting due to our relatively small sample size of training data we use cross-validation.

Figure 4.6 shows the different features of the SVM on the y-axis and their weights, normalized by the maximum value, on the x-axis. For our test, we can observe that the mean consideration time is the most important feature in the classification process. It is likely that the time the participants spend on finding the right answers is a good indicator for the workers diligence. Also the time the worker spends on answering the last question is a good indicator for the overall quality. This behavior might indicate that the worker works diligently even at the end of the test and implies a good work quality throughout the whole task. As already shown in previous studies, the overall completion time also offers a first indicator on the result quality.

The overall accuracy of the SVM amounts to 88.67%, the class precision for qualified candidates amounts to 93.06%. Considering our sample with 215 workers, this means that we predict that 10 candidates would be qualified that are truly non-qualified. For 18 workers, we predict that they are non-qualified, although they were qualified.

4.2 Accuracy and Costs of Group-based Validation Mechanisms

Assessing the quality of an individual worker is possible but rather complicated as we showed in the previous section. Therefore, many quality control mechanism are based on specialized workflows for aggregating submissions of workers to deduce an optimal task result. One well known example is the crowd-based image labeling game by Von Ahn and Dabbish [91]. In this game, a label is added to the picture, if at least two randomly picked users suggest the same label. Ahn and Dabbish argue that cheating is not possible due to the huge number of players. Two random players are very unlikely to know each other and, hence, are not able to collaborate. A similar approach was also used by Von Ahn et al. to implement the reCAPTCHA service [50].

Little et al. [92, 93] also evaluated different workflows for crowdsourcing tasks and show that parallel and iterative approaches can be used to increase the result quality of tasks and to reduce the completion time. Bernstein et at. [94] showed that multi-step interations can also be used to achieve high quality results for complex crowdsourcing tasks. Also complex workflows like Map-Reduce approaches [95] were successfully ported to the crowdsourcing environment. Going even one step further Kulkarni et al. [96] propose a system to crowdsource the design of workflows for crowdsourcing tasks. Dow et al. [97] suggest to integrate an interactive feedback system to encourage workers and Kittur et al. [98] concluded that coordination techniques improve the results in online cooperation tasks.

In contrast to quality assurance mechanisms that are based on assessing the reliability of individual workers, e.g., gold standard data, workflow based approaches comprise different types of tasks and also might include redundant task executions. This in turn raises the question about the trade-off between the overall costs of the workflow based approach and the resulting quality of the task results [99]. In the remainder of this section we evaluate this trade-off in more detail for two quality assurance mechanisms, the majority decision and a control-group based approach. We first illustrate the general concept of both approaches in Section 4.2.1 and then present an analytic model to assess the reliability of both approaches in Section 4.2.2. Section 4.2.3 introduces a cost model for each approach, which is used to evaluate the suitability of them for different use cases in Section 4.2.4. The results can then be used by employers to decide which mechanism is optimal in terms of costs and quality for a given type of task. The content of this section is mainly taken from [7, 29].

4.2.1 Modeling Group-based Quality Assurance Mechanisms

In this section we focus on two common quality assurance mechanisms, the *majority decision (MD)* and an approach using a *control group (CG)* to re-check a task. First, we describe the underlying assumptions and the basic idea of the models.

Model Assumptions

We consider a Crowdsourcing platform with N workers. If a task is completed by a set of workers, some of them might not correctly understand the task or try to gain a benefit from tricking the system. We assume that these workers do not intentionally try to corrupt the task results, but rather submit results randomly chosen from the set of all possible task results. However, it is not possible to differentiate between workers who accidentally or deliberately submitted an incorrect result by solely considering the task result. It is also not relevant for the following assumptions. Therefore, we denote to workers submitting incorrect results as *cheaters* in both cases. The probability that a randomly chosen tasks results is correct is $1 - p_{w|c}$, the probability that it is wrong $p_{w|c}$. As only

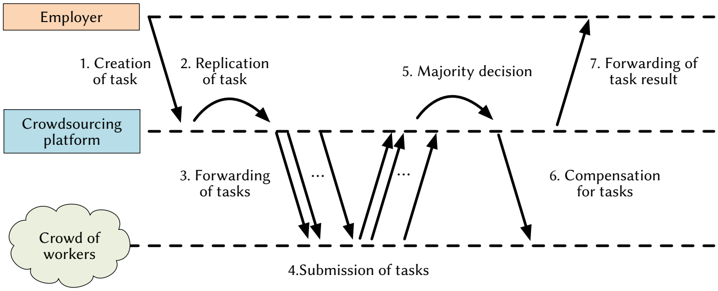


Figure 4.7: Majority Decision (MD) approach scheme.

cheaters submit incorrect results in our model, the probably of a non-cheater submitting an incorrect result is $p_{w|\overline{c}} = 0$.

The probability that a randomly chosen worker is a cheater is p_c leading to an overall probability of a wrong task result of $p_w = p_c \cdot p_{w|c}$. To illustrate this we have a look at a multiple choice test with one correct answer out of five possibilities and a crowd of 100 workers including 10 cheaters. The probability of choosing a cheater is $p_c = 10\%$, the probability for picking a wrong answer when choosing randomly is $p_{w|c} = 80\%$. This results in a probability for a wrong answer $p_w = 8\%$.

Majority Decision Approach

The first approach (MD) uses a majority decision to eliminate incorrect results and comprises the following steps illustrated in Figure 4.7. First, the employer creates a new task (1) on the Crowdsourcing platform, which is automatically replicated N_{md} times (2). The replicated tasks are forwarded to N_{md} different workers in the crowd (3), which complete the tasks and submit independent results for each of the replications (4). The platform aggregates the results by performing a majority decision (5), i.e., the result most of the workers submitted is considered to be correct. Depending on a worker's submitted result and the

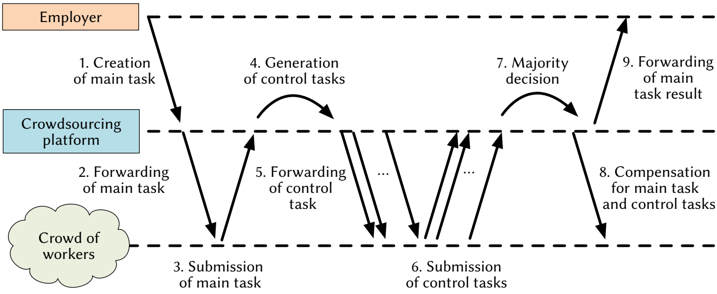

Employer
Crowdsourcing platform
Crowd of workers
1. Creation of main task
2. Forwarding of main task
3. Submission of main task
4. Generation of control tasks
5. Forwarding of control task
...
...
6. Submission of control tasks
7. Majority decision
8. Compensation for main task and control tasks
9. Forwarding of main task result

Figure 4.8: Control Group (CG) approach scheme.

outcome of the majority decision, different payouts are given to the workers (6). The submission which was chosen by the majority of the workers is assumed to be correct and forwarded to the employers (7).

A possible application for the MD approach are, e.g., object recognition tasks. In these tasks, every workers submits a binary result, whether an image contains a certain object or not. A correct labeling of the image is possible even if some workers submit incorrect results as long as the majority of the workers tags the image correctly.

The Control Group Approach

Our second approach (CG) is based on the use of a control group and is schematically depicted in Figure 4.8. The employer creates the main task on the crowdsourcing platform (1) and the task is directly forwarded to the crowd (2). Only one worker completes this main task and submits a result (3). As soon as the result for the main task is received, the crowdsourcing platform generates N_{cg} control tasks (4). Each control tasks aims at obtaining one voting on whether the main task has been performed correctly or not. The control tasks are then again forwarded to the crowd (5) and completed by a group of N_{cg} workers (6) different from the worker completing the main task. Again, a majority decision is performed on the results of the control tasks (7) and the main task is considered to be correct, if the majority of the control group workers voted accordingly. The worker completing the main task is paid depending on the rating of the majority decisions, the worker in the control group depending on their individual voting and the result of the majority decision (8). If the main task is rated valid, the result is returned to the employer (8). Otherwise, the main task is repeated by another worker until the first result is rated valid by the control crowd. This step is left out in the illustration for the sake of readability. An important point of this approach is that the main task and the "re-check" task are assumed to have different costs. Usually, the main task is expensive, while the control task is cheaper, due to the different levels of complexity.

One possible application of this approach is content creation task. Here, a worker is supposed to write a text, e.g., based on given keywords. This task is rather complex, time consuming, and thus also quite expensive. In order to assess the quality of the content, the text is given to a control group which judges whether it matches the given set of keywords or not. Compared to the original writing tasks, these tasks are easier and consequently also less expensive. Based on the majority rating of the control group, the text is accepted or rejected.

4.2.2 Evaluation of the MD and the CG Approach

Both the MD and CG approach use a majority decision as building block in order to verify the task results. In the following we have a closer look at the number N_m of workers used for this majority decision and how to optimize it, to minimize the costs and maximize the reliability of the results. Afterwards, we evaluate the probability of obtaining correct results for both approaches.

Group Size for Majority Decisions

We use N_m random workers from the total crowd of N workers for the majority decision building block and obtain N_m independent results. Each of the N_m results is incorrect with a probability of p_w. Thus, the number of incorrect results

X follows a binomial distribution $X \sim \text{BINOM}(N_m, p_w)$. To obtain a correct majority decision, the number of incorrect results has to be smaller than ${}^{N_m}/_2$, i.e., the probability of a correct majority decision p_m is given by

$$p_m(N_m) = P\left(X < \tfrac{N_m}{2}\right) = \sum_{k=0}^{\left\lfloor \frac{N_m - 1}{2} \right\rfloor} \binom{N_m}{k} p_w^k (1 - p_w)^{N_m - k}, \qquad (4.1)$$

which is depending on the probability of a wrong task result p_w, the group size N_m, and also the parity of the group, as we show in the following.

Assuming an even number $2n, n \in \mathbb{N}$ of workers which participate in the majority decision. The maximal number of incorrect results, which still leads to a correct majority decision is $2n/2 - 1 = n - 1$. Therefore, the probability for a correct majority decision of an even group is

$$p_{m_e}(2n) = P(X \leq n - 1) = \sum_{k=0}^{n-1} \binom{2n}{k} p_w^k (1 - p_w)^{2n-k}. \qquad (4.2)$$

Assuming an odd number $2n - 1, n \in \mathbb{N}$ of workers which participate in the majority decision, the maximal number of incorrect results still leading to a correct majority decision is $\lfloor (2n-1)/2 \rfloor = \lfloor n - 1/2 \rfloor = n - 1$. Hence, the probability for a correct majority decision of an odd group is

$$p_{m_o}(2n - 1) = P(X \leq n - 1) = \sum_{k=0}^{n-1} \binom{2n-1}{k} p_w^k (1 - p_w)^{2n-1-k}. \qquad (4.3)$$

Using Equation 4.3 and Equation 4.2 we show that using smaller groups of odd parity yields the same or better results than an even group with one more worker, for $n \geq 2$.

Theorem. *The probability of a correct majority decision p_{m_o} using an odd group of $2n - 1$ workers is equal to or greater than the probability of a correct majority decision p_{m_e} using an even group of $2n$ workers, for $n \in \mathbb{N} \wedge n \geq 2$.*

Proof. **Base case** $n = 2$**:** The probability for a correct majority decision using an even number of $2n = 4$ workers is

$$\begin{aligned} p_{m_e}(2n) &= P(X \leq n-1) & &= P(X \leq 1) \\ &= \sum_{k=0}^{n-1} \binom{2n}{k} p_w^k (1-p_w)^{2n-k} & &= \sum_{k=0}^{1} \binom{4}{k} p_w^k (1-p_w)^{4-k} \\ &= -3 \cdot {p_w}^4 + 8 \cdot {p_w}^3 - 6 \cdot {p_w}^2 + 1. \end{aligned}$$

The probability for a correct majority decision using an odd number of $2n - 1 = 3$ workers is

$$\begin{aligned} p_{m_o}(2n-1) &= P(X \leq n-1) & &= P(X \leq 1) \\ &= \sum_{k=0}^{n-1} \binom{2n-1}{k} p_w^k (1-p_w)^{2n-k} & &= \sum_{k=0}^{1} \binom{3}{k} p_w^k (1-p_w)^{3-k} \\ &= 2 \cdot {p_w}^3 - 3 \cdot {p_w}^2 + 1. \end{aligned}$$

The difference between $p_{m_o}(2n-1)$ and $p_{m_e}(2n)$ is

$$\begin{aligned} p_{m_o}(2n-1) - p_{m_e}(2n) &= (2 \cdot {p_w}^3 - 3 \cdot {p_w}^2 + 1) \\ &\quad - (-3 \cdot {p_w}^4 + 8 \cdot {p_w}^3 - 6 \cdot {p_w}^2 + 1) \\ &= 3 \cdot p_w^4 - 6 \cdot p_w^3 + 3 \cdot p_w^2 \\ &= 3 \cdot p_w^2 (p_w^2 - 2 \cdot p_w + 1) \\ &= 3 \cdot p_w^2 (p_w - 1)^2 \geq 0, \text{ with } 0 \leq p_w \leq 1. \end{aligned}$$

Induction hypothesis: Suppose the theorem holds for all values up to $n \in \mathbb{N}$.

$$p_{m_o}(2n-1) \geq p_{m_e}(2n)$$
$$\sum_{k=0}^{n-1} \binom{2n-1}{k} p_w^k (1-p_w)^{2n-1-k} \geq \sum_{k=0}^{n-1} \binom{2n}{k} p_w^k (1-p_w)^{2n-k}$$

Induction step: $n \rightarrow n+1$:

$$\begin{aligned}
p_{m_o}(2(n+1)-1) &= P(X \leq (n+1)-1) \\
&= \sum_{k=0}^{(n+1)-1} \binom{2(n+1)-1}{k} p_w^k (1-p_w)^{2(n+1)-1-k} \\
\text{substitute: } m = n+1 \quad &= \sum_{k=0}^{m-1} \binom{2m-1}{k} p_w^k (1-p_w)^{2m-1-k} \\
\text{Induction hypothesis} \quad &\geq \sum_{k=0}^{m-1} \binom{2m}{k} p_w^k (1-p_w)^{2m-k} \\
\text{resubstitute} \quad &= \sum_{k=0}^{(n+1)-1} \binom{2(n+1)}{k} p_w^k (1-p_w)^{2(n+1)-k} \\
&= p_{m_e}(2(n+1)) \qquad \square
\end{aligned}$$

Using Equation 4.2 and Equation 4.3 we can numerically evaluate how many workers are needed for the 99% quantile of a correct majority decision using a group with even or odd parity for different values of p_w. The results are depicted in Figure 4.9. Note that the y-axis is in logarithmic scale.

From Figure 4.9 we can observe two findings. First, we can at least save one worker when using an odd instead of an even group size without reducing the quality of the majority decision. Second, the higher p_w is, the more workers can be saved using an odd group size. An intuitive explanation for the observed effects are possible draws that occur using an even group size. In the previous

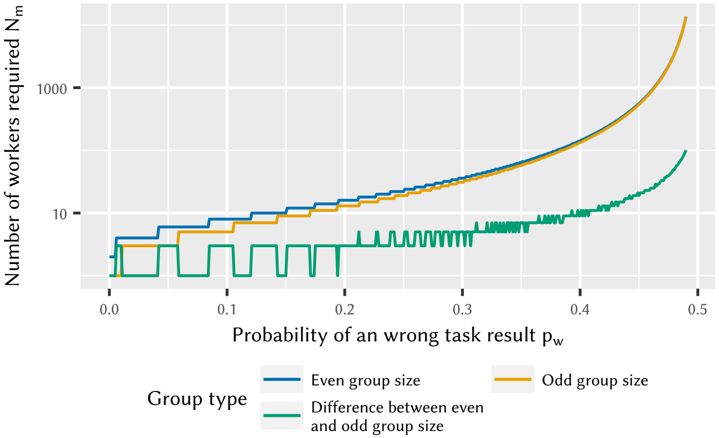


Figure 4.9: 99% Quantile of the required number N_m of workers for a correct majority decision depending on the probability of a wrong task result p_w.

considerations the draw events were also subsumed as incorrect majority decision result as no final conclusions could be drawn from this result. To avoid these effects we only use odd group sizes for majority decisions in the remainder of this work.

Quality Comparison of MD and CG Approach

Next we compare the MD and the CG approach with respect to their capability of detecting invalid results. To compare both approaches, we use the same number of workers N_m for the MD approach and for the control group of the CG approach, i.e. $N_{md} = N_{cg} = N_m$. Further, to improve the readability we use p_m instead of $p_m(N_m)$.

Having a look at the MD approach, we need to evaluate whether the group made a correct or an incorrect decision. The probability for a correct result using the MD approach p_{md} is the same as the one given in Equation 4.1. Thus,

$$p_{md} = p_m. \tag{4.4}$$

The probability for an incorrect MD result $\overline{p_{md}}$ is given by

$$\overline{p_{md}} = 1 - p_m. \tag{4.5}$$

For the CG approach we have to differentiate if the worker's and the control group's results are correct or not. This results in four possible cases are:

		Main worker	
		Correct result	*Incorrect result*
Control group	*Correct decision*	Correct task approved (CA)	Incorrect task disapproved ($\overline{CA}$)
	Incorrect decision	Correct task disapproved ($C\overline{A}$)	Incorrect task approved ($\overline{C}A$)

We assume that our crowd is very large, thus the main worker and the workers from the control group do not know each other. Hence, the cheating probability of the main worker p_w and the probability of a wrong control group decision $1 - p_m$ are independent and the probabilities of the possible outcomes of the CG approach are given by:

		Main worker	
		Correct result	*Incorrect result*
Control group	*Correct decision*	$P(CA) = p_{CA} = (1 - p_w) \cdot p_m$	$P(\overline{CA}) = p_{\overline{CA}} = p_w \cdot p_m$
	Incorrect decision	$P(C\overline{A}) = p_{C\overline{A}} = (1 - p_w) \cdot (1 - p_m)$	$P(\overline{C}A) = p_{\overline{C}A} = p_w \cdot (1 - p_m)$

Consequently, the probability for a correct result using the CG approach p_{CG} is hence given by

$$p_{cg} = p_{CA} + p_{\overline{CA}} = p_m, \tag{4.6}$$

and the probability for an incorrect result using the CG approach $\overline{p_{\text{CG}}}$ by

$$\overline{p_{cg}} = p_{C\overline{A}} + p_{\overline{C}A} = 1 - p_m. \tag{4.7}$$

Comparing p_{md} and p_{cg}, we see that both the MD and the CG approach have the same probability of producing a correct result

$$p_{md} = p_{cg} = p_m, \tag{4.8}$$

but they differer among their applicability for different crowdsourcing tasks and their costs, as we show in the next sections.

4.2.3 Cost Model for the MD and CG approach

The presented techniques are intended to be used in real crowdsourcing applications, thus the economic aspect is important and has to be considered. To this end, we develop a cost model for both approaches in the following.

Cost Model

Each worker who submits a correct task result is paid c_c, each worker submitting an incorrect result is paid c_w. In general $c_c \gg c_w$ and often incorrect submissions are not paid at all, i.e., $c_w = 0$. Approving an invalid task does not only waste money, but has further negative impacts, e.g., encouraging workers to cheat. To account for these negative effects, we introduce costs c_{fp} for a "false-positive approval", if an invalid task is not detected. Not paying for correct work has negative influences, too, as workers stop working for this employer. Hence, we use a penalty c_{fn} for a "false-negative approval", if a correct task is assumed to be invalid. As mentioned above, the control task in the CG approach is usually easier than the main task and paid differently. Thus, we use different costs for the control tasks, c_{cc},c_{cw}, c_{cfp}, and c_{cfn}, which we assume to be lower than their corresponding costs from the main task.

We now calculate the expected costs for both approaches. We use $N_{md} = N_{cg} = N_m$ workers, consequently, the probability for a correct MD and CG approach result is p_m. This analysis helps employers to decide, which approach is cheaper for a given use cases.

First we consider the MD approach. When performing a majority decision using N_{md} workers, we receive N_{mdc} correct results and $N_{md\bar{c}}$ wrong results from the workers, with

$$0 \leq N_{mdc} \leq N_{md},\ 0 \leq N_{md\bar{c}} \leq N_{md}, \text{ and } N_{md} = N_{mdc} + N_{md\bar{c}}.$$

As we use odd group sizes, the MD approach always returns a result which is assumed to be correct.

If the majority decision is correct ($N_{md\bar{c}} < {}^{N_{md}}/_{2}$), the workers who submitted correct results are paid c_c, the workers who submitted wrong results are paid c_w. However, if the majority of the workers submits a wrong result ($N_{md\bar{c}} \geq {}^{N_{md}}/_{2}$), this result is assumed to be correct. Thus, the workers who submitted wrong results are paid c_c and each worker who submitted a correct result is paid c_w. In this case there are also additional costs for the false positive approval of the task and the rejection of the correct ones. This results in the conditional costs $C_{md,N_{md\bar{c}}}$ for $N_{md\bar{c}}$ wrong results, with

$$C_{md,N_{md\bar{c}}} = \begin{cases} N_{md\bar{c}} \cdot c_w + N_{mdc} \cdot c_c, & N_{md\bar{c}} < {}^{N_{md}}/_{2} \\ N_{md\bar{c}} \cdot (c_c + c_{fp}) + N_{mdc} \cdot (c_w + c_{fn}), & N_{md\bar{c}} \geq {}^{N_{md}}/_{2}. \end{cases} \tag{4.9}$$

Using the conditional costs $C_{md,N_{md\bar{c}}}$, we can now calculate the expected costs $E[c_{md}](N_{md}, p_w)$ of the MD approach in dependency of the number of workers N_{md} involved and the probability p_w of a wrong task result. For sake of readability, we use c_{md} instead of $E[c_{md}]$.

$$
\begin{aligned}
c_{md} &= \sum_{i=0}^{m} C_{md,N_{md\bar{c}}} \cdot P(N_{md\bar{c}} = i) \\
&= \sum_{i=0}^{\left\lfloor \frac{N_{md}-1}{2} \right\rfloor} C_{md,N_{md\bar{c}}} \cdot P(N_{md\bar{c}} = i) \\
&\quad + \sum_{i=\left\lceil \frac{N_{md}}{2} \right\rceil}^{N_{md}} C_{md,N_{md\bar{c}}} \cdot P(N_{md\bar{c}} = i) \\
&\overset{\text{Eq. 4.9}}{=} \sum_{i=0}^{\left\lfloor \frac{N_{md}-1}{2} \right\rfloor} (N_{md\bar{c}} \cdot c_w + N_{mdc} \cdot c_c) \cdot P(N_{md\bar{c}} = i) \\
&\quad + \sum_{i=\left\lceil \frac{N_{md}}{2} \right\rceil}^{N_{md}} (N_{md\bar{c}} \cdot (c_c + c_{fp}) + N_{mdc} \cdot (c_w + c_{fn})) \cdot P(N_{md\bar{c}} = i).
\end{aligned}
\tag{4.10}
$$

Next, we have a closer look at the CG approach. In the CG approach one worker is working on the main task, which costs c_c if the worker completes it successfully, otherwise the worker is paid c_w. Recall that the main task is controlled by N_{cg} workers, which impose additional costs. Each of the N_{cg} workers who submitted the same results as the majority of the control crowd is paid c_{cc}, the rest of the workers c_{cw}. Similar to the MD approach there are penalties for approving wrong results and rejecting correct results.

The costs vary, depending on whether the worker of the main task is cheating or not, and whether the control crowd rates the result of the main task correctly. To calculate the total expected costs $E[c_{cg}]$ we have to consider four cases.

		Main worker	
		Correct result	*Incorrect result*
Control group	*Correct decision*	$c_{CA} = c_c + c_{md}$	$c_{\overline{CA}} = c_w + c_{md}$
	Incorrect decision	$c_{C\overline{A}} = c_w + c_{md} + c_{fn}$	$c_{\overline{C}A} = c_c + c_{md} + c_{fp}$

In each of the four cases, the control crowd is paid. As we use the same number of workers for the control crowd in the CG approach as for the majority decision in the MD approach ($N_m = N_{md} = N_{cg}$), the cost of the control crowd in the CG approach are calculated using Equation 4.10 and the costs c_{cc}, c_{cw}, c_{cfp}, and c_{cfn}. Now we have a closer look at the varying costs. If the main worker submits a correct result and the control crowd approves it, then there are only the additional costs c_c for the payment of the main worker. Similarly, if the main worker submits a wrong result and the control crowd realizes this and disapproves the task. In that case the worker is paid c_w. If the control crowd falsely disapproves a correct main task, the main worker is paid c_w as his work is assumed to be incorrect and a penalty of c_{fn} is added. If the control approves an incorrect task, the main worker is paid c_c for his result and a penalty c_{fp} is added for the incorrect approval.

In the CG approach, the main task is repeated until the control crowd rates it to be correct. This happens, if a correct main task is approved (CA) or if a wrong task is approved ($\overline{C}A$). Thus, the probability $P(cg_{\text{approve}})$ of approving the main task is

$$P(cg_{\text{approve}}) = P(CA \cup \overline{C}A) = p_{CA} + p_{\overline{C}A} = p_m + p_w - 2 \cdot p_m p_w,$$

and the number of repetitions R until the main task is approved follows a geometrical distribution

$$P(R = n) = P(cg_{\text{approve}}) \cdot (1 - P(cg_{\text{approve}}))^{(n-1)}.$$

The expected costs $E[c_{cg}]$ of the CG approach consist of $E[R]-1$ times the cost $c_{disapproval}$ for disapproving the main task and once the costs $c_{approval}$ for approving the main task. Both costs are again depending on whether the control crowd made a correct decision or not.

$$
\begin{aligned}
E[c_{approval}] &= P(CA|cg_{\text{approve}}) \cdot c_{CA} + P(\overline{C}A|cg_{\text{approve}}) \cdot c_{\overline{C}A} \\
E[c_{disapproval}] &= P(C\overline{A}|cg_{\text{disapprove}}) \cdot c_{C\overline{A}} + P(\overline{CA}|cg_{\text{disapprove}}) \cdot c_{\overline{CA}}.
\end{aligned}
$$

The expected costs $E[c_{cg}]$ of the CG approach can now be calculated as follows. For sake or readability, we use R instead of $E[R]$, c_{cg} instead of $E[c_{cg}]$, $c_{approval}$ instead of $E[c_{approval}]$, and $c_{disapproval}$ instead of $E[c_{disapproval}]$.

$$
\begin{aligned}
c_{cg} =& c_{approval} + (R-1) \cdot c_{disapproval} \\
=& \frac{p_{CA}}{P(cg_{\text{approve}})} \cdot c_{CA} + \frac{p_{\overline{C}A}}{P(cg_{\text{approve}})} \cdot c_{\overline{C}A} + \\
& (R-1) \cdot \left(\frac{p_{C\overline{A}}}{P(cg_{\text{disapprove}})} \cdot c_{C\overline{A}} + \frac{p_{\overline{CA}}}{P(cg_{\text{disapprove}})} \cdot c_{\overline{CA}} \right).
\end{aligned}
\tag{4.11}
$$

After developing a cost model for the MD and CG approach, we now evaluate the costs of both approaches for different exemplary task types.

Impact of the different costs factors of the MD and CG approach

The presented cost model includes different cost factors which, depending on p_w, contribute more or less to the total costs of the approach. In the following we have a closer look at the composition of the overall costs depending on p_w. Figure 4.10 shows the total costs of the MD approach depending on p_w and the contribution of the individual cost factors to the total costs. c_c, c_w, c_{fp}, and c_{fn} are all set to 1 and N_{md} =11 workers are used.

First we have a look at the total costs. In this example, the minimal costs are 11 for $p_w = 0$. Here, all N_{md} =11 workers submit the correct result and are

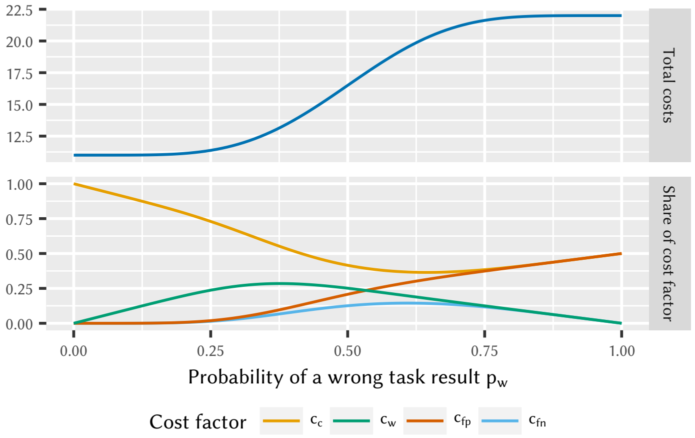


Figure 4.10: Cost factors of the MD approach.

therefore paid $c_c = 1$. Further not additional costs occur here. With increasing p_w also the total costs increase, with a maximums at $p_w = 1$. This results from all workers being paid the same amount no matter if they vote according the majority or not (c_c =c_w =1) and the increasing probability for the penalties c_{fp} and c_{fn} with the increase of p_w. For different values of c_c, c_w, c_{fp}, and c_{fn}, the costs c_{md} might become minimal for $p_w \neq 0$.

Next, we focus on the contributions of the different cost factors depending on p_w. Note that the contributions of the cost factors have been normalized, so that to the sum of the cost factors equals to one for all p_w. First, we have a look at c_c. As the probability for a wrong answer p_w is zero, all workers are paid c_c, consequently the total costs are only influenced by c_c With the increase of p_w, some workers are submitting wrong results and are no longer paid c_c, but other cost factors start contributing to the total costs. Thus, the share of c_c with respect to the total costs decreases. For, large values of p_w, the wrong results are no longer detected and again more workers are paid c_c, even if they sub-

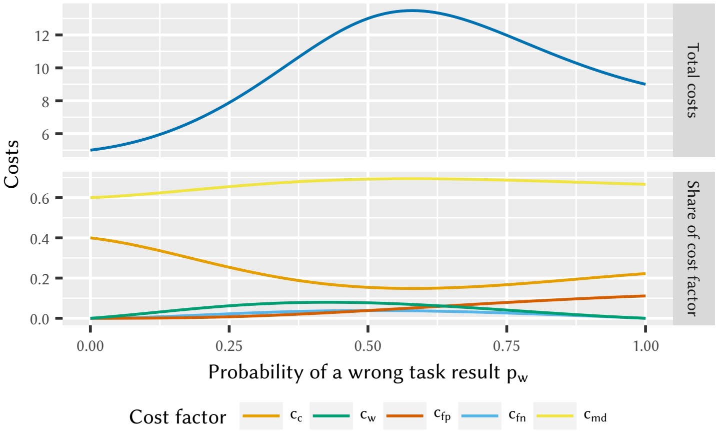


Figure 4.11: Cost factors of the CG approach.

mitted invalid results. Consequently, the impact of c_c again slightly increases. c_w is complementary to c_c as each worker who is not paid c_c receives c_w. With p_w increasing, also the probability of a wrong majority decision increases. This results in a larger contribution of c_{fp}, as wrong answers are more likely to be assumed correct. This is similar for c_{fn}. As p_w increases, more correct results are assumed to be incorrect and the contribution of c_{fn} increases. However, with increasing p_w also the number of correct answers decreases and consequently the impact of c_{fn} decreases again for large values of p_w.

Figure 4.11 shows the expected costs of the CG approach depending on p_w and the contribution of the individual cost factors. c_w, c_{fp}, and c_{fn} are set to 1 and N_{cg} =3 workers are used for the control task with $c_{cc} = c_{cw} = c_{cfp} = c_{cfn} = 1$. The main task should be paid higher than the control task, thus we set c_c =2.

Again we first focus on the total costs of the approach. Similar to the MD approach, the lowest costs can be observed for $p_w = 0$. Here, the main task has to be performed only once and also the control group correctly approves the task. For increasing values of p_w also the costs increase, however in contrast to the MD approach, a maximum is reached for $p_w \neq 1$. Thereafter, the costs decrease again. This can be explained by the fact that with increasing p_w it becomes more likely to detect an invalid main task. In this case, the main task has to be repeated and another majority decision has to be performed by the control group leading to additional costs. As p_w increases further, the control group becomes more likely to no longer detect invalid task submissions. This in turn again decreases the number of repetitions until main task submission is approved, and consequently the number of performed majority decisions including the resulting costs.

This can also be observe when having a look at the contribution of the individual cost factors. We see that the impact of c_{md} first increases with p_w and later slightly decreases again for very high values of p_w. c_c has the highest impact for low values of p_w, however, its impact decreases for higher values of p_w, since here the repeated majority decisions of the control group become a larger cost influencing factor. Similar to the MD approach, the impact of c_w is complementary to c_c as every main worker is either paid c_c or c_w. c_{fp} and c_{fn} behave similar than for the MD approach.

4.2.4 Identification of Optimal Validation Mechanisms

In the following we use the developed cost modes to identify the cost-optimal approach for different kinds of crowdsourcing tasks, i.e. routine, complex and creative tasks.

Routine Tasks

Routine tasks are typically low paid with $c_c = 1$ for the main task. The task of re-checking the main task should not be higher paid, thus, $c_{cc} = 1$ for the control task. Workers submitting wrong results are generally not paid, thus $c_w = c_{cw} = 0$. The costs caused by not detected cheating workers are very low in this case, but as the worker might be encouraged to continue cheating

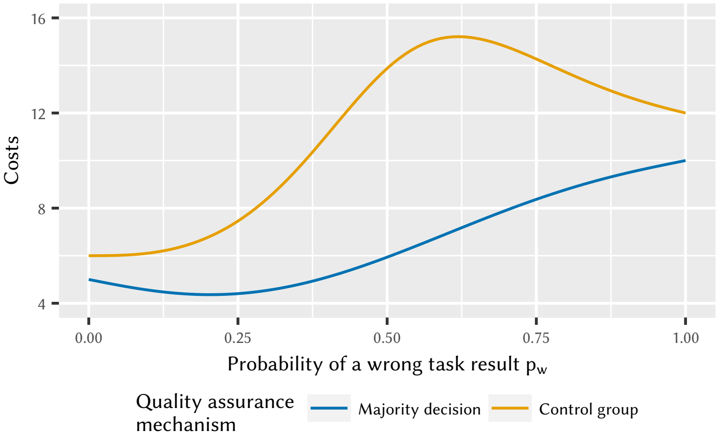


Figure 4.12: Costs of a routine task dependent on the probability of a wrong task result p_w.

we impose a penalty for each approval of an invalid task of $c_{fp} = 1$. Refusing to pay a worker who completed his task, will stop the worker from working for this employer. However, the crowd contains many workers who can complete simple tasks, the penalty $c_{fn} = 1$ is low. For the control crowd we also use the same penalties $c_{cfp} = c_{cfn} = 1$. The group size for the majority decision building block in both approaches is $N_{md} = N_{cg} = 5$.

The resulting costs depending on p_w for both approaches are shown in Figure 4.12. For low values of p_w, c_{md} slightly decreases as the workers voting against the majority are no longer paid. With a further increase of p_w, we see the same effects as shown in Figure 4.10, without the contribution of c_w. The development of c_{cg} differs slightly from the one depicted in Figure 4.11, as $c_{cw} = c_w = 0$ and $N_{cg} = 5$ but shows the same effects. Due to the larger control group here, the costs increase faster if the main task and the control groups decisions have to be repeated leading to a sharper maximum.

For very small and very large values of p_w the costs c_{md} and c_{cg} differ only a bit. Still, in the case of a routine task, the costs of the CG approach are always higher than the costs of the MD approach, since $c_{cc} = c_c$ and $c_{fp} \approx c_c$. Thus, the MD approach should be preferred for routine tasks.

Complex and Creative Tasks

Complex and creative tasks are usually higher paid then routine tasks due the higher skill requirements. Thus, we set $c_c = 5$ in this case. However, workers producing results with minor quality are also not paid for this kind of tasks, i.e., $c_w = 0$. Even if the main task is complex, checking the quality of the tasks should be more simple and consequently the reward for the control group workers is smaller then the reward for the main tasks. In our example, each member of the control group is paid $c_{cc} = 1$ if he rates according to the majority, otherwise he is paid $c_{cw} = 0$. Accepting invalid main tasks can significantly affect the employer for these kind of tasks, thus the penalty for approving low quality texts is very high $c_{fp} = 20$. Besides this, qualified workers for the main task are rare and losing one of them is not desirable. Therefore, we assume $c_{fn} = 5$. The control crowd workers do not require special qualifications and a few miss-ratings can be tolerated, thus we choose $c_{cfn} = c_{cfp} = 1$. The group size for both approaches is again $N_{md} = N_{cg} = 5$.

The resulting costs are depicted in Figure 4.13. c_{md} and c_{cg} show a similar shape to that for the routine task. However, observe that in this case the CG approach with $c_{cc} = 1$ is always cheaper than the MD approach, since the penalty for false positive approvals is high and the costs for the control task $c_{cc} < c_c$ are low. If the costs for the control task are raised to $c_{cc} = c_c = 5$, the CG approach only performs better than MD for $p_w > 0.52$.

Using these results we derive a guideline for complex and creative tasks. If the cost ratio $^{c_{cc}}/_{c_c} \ll 1$ holds, the CG approach should be favored. Otherwise, a more detailed analysis is required. Therefore, we have a look at the impact of c_{fp} and c_{fn} on the choice of the optimal validation approach.

For this analysis we use N_{md} = N_{cg} =5 and normalize all costs to c_c. c_{cc} is

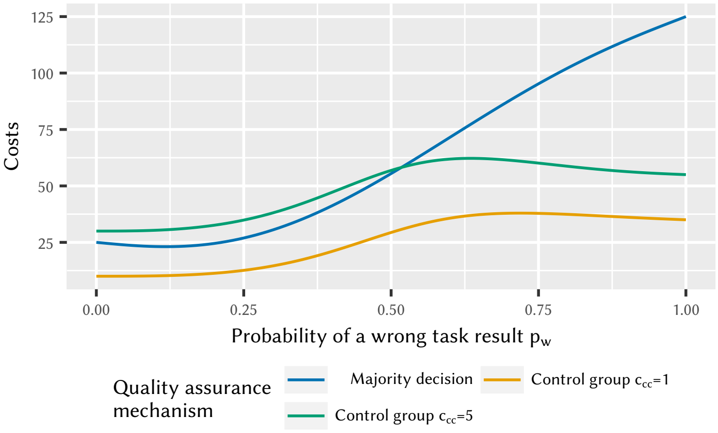


Figure 4.13: Costs of a complex or creative task depending on the probability of a wrong task result p_w

generally smaller than c_c, thus we choose $c_{cc} = {}^{c_c}/_2$ for this analysis. As a few miss-ratings of the control-crowd are tolerable we use c_{cfp} =c_{cfn} =c_{cc}. Similar to the previous example we set c_w = c_{cw} = 0. In order to analyze the impact of c_{fp} and c_{fn} we vary both penalties from c_c to $10 \cdot c_c$ and determine the costs for both approaches for different values of p_w.

Figure 4.14 visualizes the cost-optimal approach depending on c_{fp} and c_{fn} for four exemplary values of p_w. The x-axis shows the values of c_{fn} and the y-axis shows the values of c_{fp}. Both are normalized to c_c. The colored areas indicate the cost optimal approach for each parameter setting. It has to be noted that borders of the colored areas being a stair function as numerical inaccuracies occure here.

Consider the example of $p_w = 0.4$. If the penalty for approving an invalid task c_{fp} is higher than $2.5c_c$, which is often the case for complex and creative tasks, the CG approach is always superior to the MD approach in terms of the

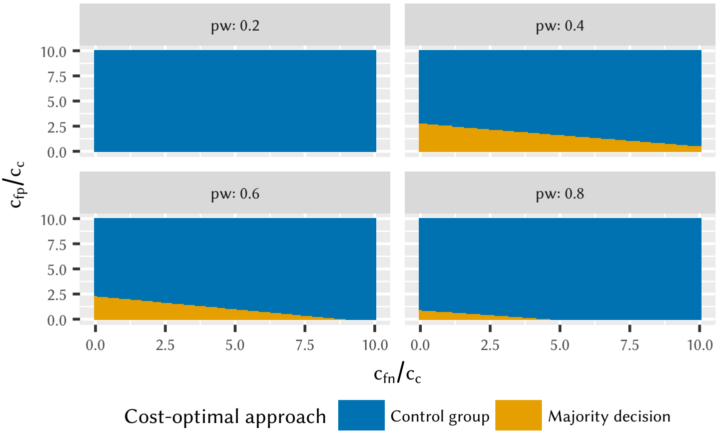


Figure 4.14: Cost-optimal approach in dependency of different penalties for false positive c_{fp} and false negative c_{fn} approvals.

costs. In case the penalty for approving an invalid task c_{fp} and also the penalty for rejecting a valid one c_{fn} are rather low ($c_{fp} < c_c$ and $c_{fn} < c_c$), which often applies to routine tasks, the MD approach should be used. In general the results depicted in Figure 4.14 show that for $^{c_{cc}}/_{c_c} \not\ll 1$, p_w, c_{fp}, and c_{fn} have to be considered while choosing the cost-optimal validation approach. However, if p_w, c_{fp}, and c_{fn} are known, the cost optimal approach can be determined numerically using Equation 4.10 and Equation 4.11.

Cost-Quality Optimization Guidelines for Complex and Creative Tasks

Finding a trade-off between quality and the costs for complex and creative tasks is important as they are in general more expensive than routine tasks. We saw previously that the CG approach outperforms the MD approach in terms of costs in most cases. Hence, we will focus on the CG approach in the following.

In order to reduce the total costs c_{cg}, a smaller control crowd can be used.

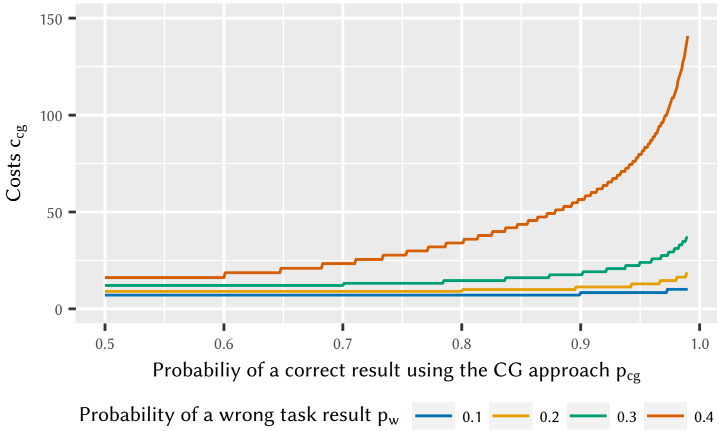


Figure 4.15: Total costs c_{cg} depending the probability of a correct result using the CG approach p_{cg} and the probability of a wrong task result p_w.

However this negatively affects the probability of obtaining a correct result as p_{cg} decreases with the group size. Consequently, a trade-off between c_{cg} and p_{cg} exists. For our evaluation we use the previous example with $c_c = 5$, $c_w = 0$, $c_{fp} = 20$, $c_{fn} = 5$,$c_{cc} = 1$,$c_{cw} = 0$,$c_{cfp} = 1$, $c_{cfn} = 1$, and $N_{cg} = 5$.

Figure 4.15 depicts the costs of the CG approach c_{cg} depending on the probability of a correct result p_{cg} for different values of p_w. Our analysis showed that c_{cg} remains almost constant for $p_{cg} < 0.5$, therefore, we focus only on $p_{cg} \geq 0.5$. We observe that c_{cg} increases with p_w and p_{cg}. A higher probability of a correct task result p_{cg} needs more workers leading to higher costs. Also with an increase of p_w more workers are required to achieve a valid result. For a small value of p_w the influence of p_{cg} on the costs is only marginal and increasing the probability of obtaining a correct result is rather cheap. For high values of p_w the costs increase tremendously with p_{cg}, which makes an detection improvement extremely expensive.

We can assume that an employer can approximately determine p_w based on the results of previous tasks. Hence, our model allows him to make a trade-off between costs and result quality according to his needs, by calculating the required N_m. To illustrate this, we have a look at two examples with $p_w = 0.4$. Assume an employer wants to spend $c_{cg} = 30$ for the campaign. We can derive from Figure 4.15 that p_{cg} will be about 0.77. For $p_w = 0.4$ Equation 4.1 can be solved numerically to $N_{cg} = \frac{ln(1-p_{cg})+1.0404}{-0.0310}$ and we can calculate the required control group size $N_{cg} = 25$. The second use case is an employer who demands $p_{cg} = 90\%$ for his campaign. We can calculate the required group size $N_{cg} = 40$ and derive $c_{cg} \approx 57$ from Figure 4.15.

Another approach for saving costs is using well trained workers, i.e., reducing p_w. To analyze the impact of trained workers on c_{cg}, we determine the 99% quantiles of the required number N_{cg} of workers for a correct decision depending on p_w. Using these group sizes, we calculate the costs of the CG approach and the savings as difference between the calculated values and the costs for the CG approach for $p_w = 0.4$. The cost savings normalized by the cost of the CG approach for $p_w = 0.4$ are shown in Figure 4.16.

We observe that using optimal workers $p_w = 0$ can save about 95% of the costs, compared to a group of workers with $p_w = 0.4$. It also show that even groups with larger values of p_w still reduce the costs significantly. This results from the huge amount of workers required for the control group if p_w increases. In general, well trained workers are more expensive than untrained workers. However, due to the large cost saving potential of these trained workers, they might nevertheless be more cost effective than untrained ones.

Creative and complex tasks require special skills. In order to attract skilled workers, these tasks are better paid than routine tasks. But the costs c_{cg} for a task using the CG approach are split between the main task worker and the control group workers, with the control group being a significant cost factor as shown in Figure 4.11. Thus, there is a trade-off between the available money for the main worker and the result quality.

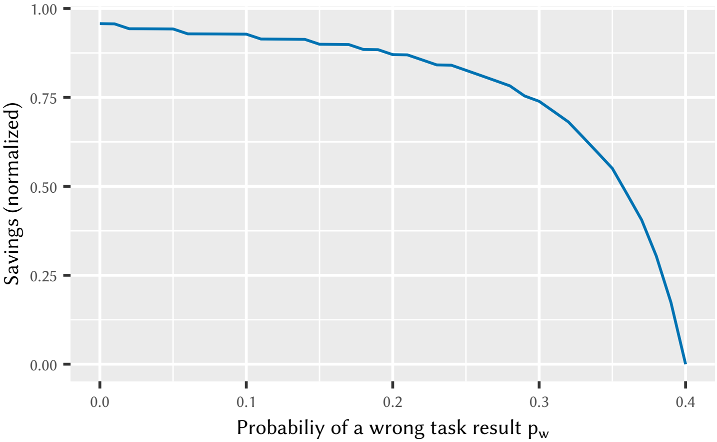


Figure 4.16: Savings due to the usage of well-trained workers.

To analyse this trade-off, we define the overhead costs $c_{\text{cg-overhead}}$ for the CG approach, which include all costs except the reward for the correct main task c_c. Consequently, for a fixed budget b, the maximal available salary c_c can be calculated by,

$$c_c = b - c_{\text{cg-overhead}}.$$

The overhead costs $c_{\text{cg-overhead}}$ can be determined by using the cost model of the CG approach and setting c_c =0.

For the further evaluation, we introduce the quotient $\varepsilon = {}^{c_c}/_{b}$ as a measure for the efficiency of the cost distribution, i.e., how much of the total costs can be invested in the reward for the main task. $\varepsilon = 1$ means that the entire budget is spent for the main task. Figure 4.17 depicts ε for different budgets b and different probabilities of a correct control group decision p_{cg}.

The intersection of the curves and the x-axis marks the minimum required task budget for the given p_{cg}. At this intersection point, no salary for the main

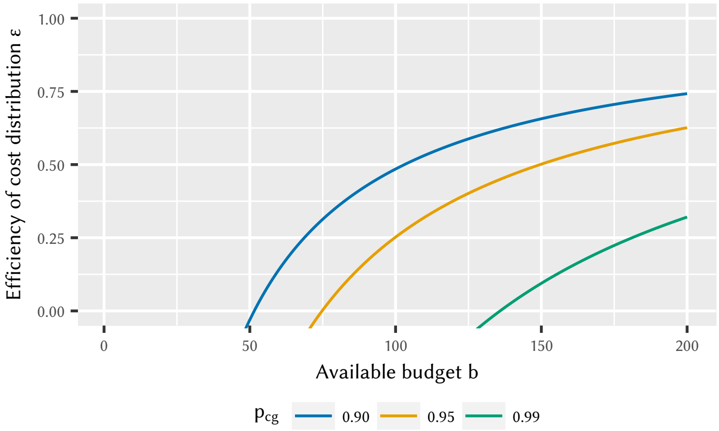


Figure 4.17: Efficiency of the cost distribution ε depending on the budget b and varying probabilities of a correct control group decision p_{cg}.

task is available. With increasing budget b, more salary for the main task is available as $c_{\text{cg-overhead}}$ remains constant. For large budgets, the main task salary is the biggest part b. The intersections of the curves and the x-axis move to the right for higher p_{cg}, which shows that the task becomes the more expensive the higher the aimed probability of a correct result is. With higher p_{cg} also the efficiency of the cost distribution degrades quickly and a large amount of the budget is spent on the control crowd instead of the main worker. Therefore, an employer has to consider carefully the required p_{cg}.

4.3 Lessons Learned

In the first part of this chapter, we showed that monitoring the worker's interactions with the task interface allows for assessing the quality of worker results. To show the suitability of this approach, we implemented a web-based language skill test based on texts taken from the Simple English version of Wikipedia and multiple choice questions about the texts' content aiming at literal comprehension, reorganization, and inference. The task interface was equipped with server- and client-side monitoring that can easily be integrated most of today's web-based crowdsourcing tasks and tracks the workers interactions, e.g., scrolling behavior and clicks.

In February 2013, we acquired 215 workers to participate in the test using the crowdsourcing platform Microworkers.com. Even if only 2% of the participants were native speakers, 18% of the participants achieved the maximum score in the test and 71% of the participants reached the qualification threshold that was set to 50% of the test points. Comparing the test scores and the test completion times of the workers, we derived completion times below 3 min as a first indicator for identifying low quality results in our exemplary task. However, a deeper analysis of completion times showed that additional measures are necessary as four workers completed the test offline in advance resulting in high test scores at very small completion time. Additionally many tests were completed in 3 to 25 min with highly varying test scores. All but two workers, who spend more than 25 min passed the test successfully.

To better understand the varying test scores of the submissions taking 3 to 25 min, we introduced the concepts of working phases. The working phase for a text describes the time a text is visible on the screen. The working phase for a question describes the difference between the timestamp of the first time the worker could have possibly seen the questions on a text and the timestamp of the last answer given for all questions on a text. For the analysis we defined two worker groups, qualified workers with scores of at least 50% of the maximum test score and non-qualified workers with test scores below this threshold. We showed that the qualified workers spent significantly more time in the working phases than the non-qualified workers and that the time spent by the qualified workers even increases during test. In contrast, non-qualified workers often spent less than the minimum reading time of 6 sec and minimum answering time of 25 sec in the working phases.

Further, we introduced the consideration time which is the time between the first time a worker saw a question and the time the worker changed his answer for this question for the last time. Again the qualified workers took more time to answer the questions and we observed that the qualified workers also spend more time on difficult questions, i.e., inference and reorganization, than on simple comprehension questions. The test also showed that wrong answers, which are plausible but not in the text tend to attract fast clicking workers.

Finally we showed that it is possible to classify a worker as qualified or non-qualified based on the introduced measures and the completion time. The overall accuracy of the trained support vector machine amounted to 88.67%, the class precision for qualified workers to 93.06%.

The implemented approach shows the potential of interaction monitoring as an additional method of assessing the reliability of workers. In contrast to traditional approaches, like gold standard data, this method does not impose additional work load to the workers which would result in higher costs. Furthermore, it also does not require the creation of training data sets and can be applied to any kind of task, even if it includes subjective ratings. A specific interaction model is required for every task, but Kazai et al. [100] recently showed that the behavior of experts can be used as gold standart here. Still, the task interface has to be designed appropriately. Further, it has to be guaranteed that the privacy of the workers is preserved while applying the monitoring techniques.

In the second part of the chapter we focused on workflow based quality assurance mechanisms. Here, we evaluated the trade-off between costs and quality for a majority decision where all workers perform the same task and a control group based approach where one worker performs the tasks and a group of workers assesses the quality of the task. To analyse the trade-off we developed a mathematical model for each approach that captures the probability of obtaining a correct result and the related costs. Using this model we proved that avoiding even groups of workers in majority decisions results in cost saving while keeping the same result quality. Further we showed that both approaches lead to the same result quality if the same number of workers is used for the majority decision and the control group.

The cost model for both approaches enabled us to identify the cost-optimal mechanisms for routine and complex tasks:

- If the costs of the main task and the task of the control group do not differ significantly, a majority decision is more cost effective. This is usually the case for routine tasks.
- For complex and creative tasks, the main task is often difficult, while controlling the task result is rather easy. This results in lower costs for the control group tasks compared to the main task. Here, a control group approach is better then a majority decision approach, but only if the costs differ significantly.
- If the costs for the control group task and the main task are only slightly different, the identification of the cost-optimal approach is more difficult. In this case also the costs resulting from accepting invalid results and discarding valid results have to be considered.

Finally, we showed how the model for the control group approach can be applied for finding trade-offs between the cheat detection quality and the resulting costs. It was found that individual error probabilities of the workers have a significant negative influence on the costs and that a smaller group of high qualified workers can lead to large savings despite their higher salary.

The models developed in the second part of the chapter can help employers on crowdsourcing platforms identifying cost-optimal quality assurance mechanisms for a large range of tasks. Further, the models can also be easily extended, e.g., to incorporate inhomogeneous error probabilities, or adapted to other quality assurance mechanisms like expert reviews [101].

5 Use-Cases of Crowdsourcing

Crowdsourcing is not only subject of ongoing research, e.g., for optimizing the result quality, but also became a valuable tool for researchers who also benefit from the easy and cost-effective access to a huge human workforce. This workforce enables the creation of extensive training sets for artificial intelligence research, e.g. [102, 103], the processing of a large amount of image data for astronomic [104] and medical [105] research, or conducting psychological tests [106, 107]. However, similar to commercial applications of crowdsourcing, the limitations of the crowdsourcing approach as well as the special requirements for crowdsourced tasks, e.g., additional quality control mechanisms, have to be considered while using crowdsourcing as research tool. Furthermore, existing research methodologies need to be adapted when transfered from existing test setting to the crowd.

In the remainder of this chapter we present two exemplary use cases of crowdsourcing in scientific research. First, in Section 5.1, we use crowdsourcing as a recruiting instrument to acquire users for network measurements. We illustrate that crowdsourcing based network measurements can be used complementary to traditional approaches in order to gain a broader and more realistic view on the current Internet infrastructure and its usage. To make full usage of this approach, we develop guidelines and best practices on how to conduct crowdsourcing based network measurements. In Section 5.2, we detail on an orthogonal use case, namely crowdsourcing based Quality of Experience tests. This poses different challenges as crowdsourced network measurements mostly depend on the crowdsourcing workers' devices as measurement probes. In contrast crowdsourced quality of experience tests focus on the collection of large numbers of subjective judgements. Due to the subjective nature of the workers' judgement the identification of reliable workers and the application of quality control mechanisms is challenging. Again, we illustrate the benefits of taking Quality of Experience tests to the crowd, show existing limitations, and give practical advises for a successful test design. Section 5.3 concludes this chapter. Note that, the content of this chapter is published in [1, 3].

5.1 Crowdsourcing-Based Network Measurements

Measurements are crucial to shedding light on eventual issues, supporting the understanding of arising problems, and improving system design of the current Internet infrastructure and network infrastructures in general. Such measurements must cover several technical aspects, e.g., signal strength or radio coverage on the *link layer* and topology, routing and dynamic traffic changes on the *network layer*. For optimizing the QoE as perceived by users, however, *application layer measurements* on the end user device and subjective studies at the *user level* are gaining importance. Additionally, they help to identify current and future network challenges and their effects on end users. All in all, measurement probes are required both within the network to measure technical parameters, and on the edge of the network to measure the QoE of individual users for specific applications.

Currently, coarse measurements are conducted via

1) passive observations of traffic in cooperation with Internet Service Providers (ISPs) and network operators,

2) actively running experiments in testbeds, either in isolation or connected to the public Internet, or

3) asking voluntary participants to run a measurement tool.

The value of those measurement tools is unquestionable, however, it is still not possible to cover all networking related aspects. Therefore we propose an extension of this toolset by introducing Crowdsourced Network Measurements (CNM) as an additional mean for researchers to complement the view and broaden the scope of previous techniques.

The remainder of this section is structured as follows. Section 5.1.1 reviews existing network measurement techniques. The concept of CNM is introduced in Section 5.1.2, and its advantages and challenges are discussed. Section 5.1.3 discusses different parameters considered when designing network measurements and to what extent they can be realized with the different techniques. Furthermore, a comparison of CNM and existing measurement methods is available here. Section 5.1.4 illustrates the advantages of CNM using some exemplary use cases. Practical guidelines for conducting CNM and avoiding common pitfalls are given in Section 5.1.5. The content of the following sections is taken from [1].

5.1.1 Existing Network Measurement Techniques

As mentioned earlier, network measurements are primarily conducted using existing infrastructure at an ISP, in testbeds, or with the help of voluntary participants. In the following, we detail on the basic principles of these different approaches.

Network Measurements by ISPs

ISPs have direct access to their network components, e.g., routers and Points-of-Presence, and thus they are able to gain detailed knowledge about their network. This includes entire information about the structure of the network and the traffic within the network. Measurements of application behavior are possible to a certain extent by using advanced tools that extract information from packet traces, e.g., using deep-packet inspection methodologies [108–110]. On the downside, the amount of data that must be processed causes new challenges, but sampling strategies and today's processing power allow easy scaling to several Gb/s [111]. This type of measurement allows to draw a very accurate picture of a specific part of the Internet. The ability to perform passive analysis using off-the-shelf hardware has made such measurements quite popular among the research community. Here, research focuses on novel methodologies to extract increasing amounts of valuable information from passive traces.

Distributed Testbeds

Testbeds, such as PlanetLab [112], M-LAB [113], GENI [114], and GLab [115], consist of hundreds of nodes located inside a country or spread around the globe. What is common to these testbeds is that they allow running distributed experiments in a well-specified environment that supports even complex measurement setups. In contrast to ISP measurements, testbeds offer the possibility of a broader view of the Internet, due to the distributed geographical locations and the different Internet connections of the nodes. Consequently, testing novel applications on PlanetLab has become the de facto standard in the research community. Similarly, PlanetLab is popular for running active measurements. However, a significant drawback is the limited and often special location of testbed nodes that decreases the generality of results [116].

Voluntary Participation of Internet Users

Another possible means of performing network measurements relies on voluntary participants. DIMES [117], iPlane [118], or DipZoom [119] are among the first attempts in this direction. These measurement tools have been made available to the community and volunteers asked to participate in these experiments. However they are still not widely used on residential hosts, but the majority of the participating hosts are PlanetLab nodes with some nodes from academia.

To access a broader range of end user devices, projects attempt to ease the installation of software. One technique employed is distributing plug-ins for popular software to conduct measurements, e.g., creating a Firefox browser extension [120] or distributing a plugin for BitTorrent clients [121, 122]. Another idea is providing measurement devices to end users as done by Sam-

Knows [123] or Ripe Atlas [124] to create large testbeds. Finally, applications such as Skype or the streaming solution by Conviva [125] embed network measurement tools to monitor a specific service and provide collection information for the service owner.

In contrast to paid crowdsourcing, no monetary incentives are involved here. However, a thoughtful incentive design including, e.g., the type of incentive and when to grant it is crucial for motivating a sufficient number of participants in a voluntary measurement context. Some of the projects provide incentives, e.g., access to the observed information, access to other participating measurement probes, or improvement of the participant's network performance [126]. However, these incentives primarily target interested and experienced technical users or other researchers. To reach a broader group, incentive mechanisms must be adapted depending on the required target group of participants and the actual desired measurements, causing incentive design to become a difficult challenge [3].

5.1.2 Crowdsourcing-Based Network Measurements

Our proposal to complement the existing network measurement techniques, is the use of paid crowdsourcing as an additional method to acquire results from end users. In this section, we detail on the advantages and challenges related to this technique, and the resulting strengths, weaknesses, opportunities and threads.

Reusing the categorization from Section 2.1.3, CNM tasks belong to the category of routine tasks as CNM tasks are mainly simple and often highly repetitive, e.g., generating consecutive measurement samples. The main differences between paid micro-tasking and voluntary participation are the possibility to select dedicated participants and the primarily monetary incentives in paid crowdsourcing. These are also the main reasons for some of the advantages and drawbacks of this technique, which are discussed in the following.

Promising Advantages

CNM usually results in *low costs for measurements*, even for very large-scale experiments. Almost no infrastructure is required to conduct a measurement, except for handling the reporting of the measurement probes. Only the costs for the crowdsourcing platform usage and the salary of the workers must be considered.

Another advantage is that the workers, and consequently the CNM probes, exhibit a very high *diversity*. Crowdsourcing workers are usually distributed all over the world, as shown in Section 3.1.1, allowing researchers to conduct measurements from various geographical locations and multiple ISP networks. The workers access the Internet using different types of broadband access technologies and a large variety of devices, ranging from desktop PCs to smartphones, enabling a diverse view of the network.

This variety of real end user devices allows measurements in *realistic scenarios*. Consequently, the results are not biased by special equipment or by special (high-speed) Internet connections, which are typical for research facilities. Crowdsourcing-based probes can also be instrumented to collect information about commonly used software on end user devices. Moreover, measurements on the end user device can easily involve the workers enabling large-scale, realistic *QoE measurements*.

Of course, end user measurements are also possible using voluntary participation approaches. However, CNM offers a better *controllability of the probes*. The large variety and number of crowdsourcing workers allows researchers to choose only a subset of workers suitable for meeting the specific requirements of the measurement, e.g., in terms of country of origin or hardware and software on the worker's device. Furthermore, measurement tools can be implemented to gather exactly the required level of detail of the measurement data without any additional censoring in a post-processing step.

Finally, the large number of workers on commercial crowdsourcing platforms offers a 24/7 workforce with thousands of potential measurement probes being online at the same time. This enables not only large-scale measurement campaigns but also a *rapid generation of measurement results*, with several hundreds of tasks being processed in a few hours, or even minutes.

Emerging Challenges

We saw that CNM enables several new possibilities for network measurements and it is obvious that reusing existing measurement tools and methodology is desirable. However, it can be difficult to adapt existing measurement approaches to incorporate crowdsourcing workers as several new challenges emerge.

The *diversity of end user devices*, one of the major advantages of this approach, can cause significant issues during the test setup. Namely, CNM probes will most likely differ in their Operating Systems (OSs), software and hardware configurations and their network connection. This must be considered in the design of the measurement. It may be necessary to adapt, e.g., the duration of the measurement or the amount of transferred data based on the available bandwidth of the measurement probes. The measurement software must also provide means for detecting *untrustworthy workers* and *cheaters*. Therefore, additional effort is required to add security checks to the measurement software as discussed in Chapter 4.

Another challenge that needs to be addressed during the design phase is the *coordination of the workers*, since the workers decide themselves which task they work on. This makes it difficult to schedule measurements at a very specific point in time. If further filtering of the workers is applied, e.g., selecting customers of a given ISP, the group of potential workers shrinks, and fulfilling additional constraints, e.g., a minimum number of simultaneous probes, can become a challenge.

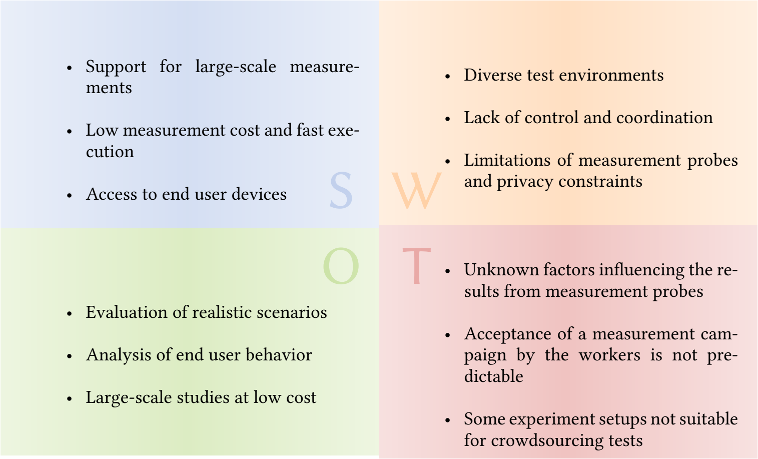
Additionally, recruiting workers from specific commercial platforms can be difficult due to *restrictions and limitations of the Crowdsourcing platforms*. Specialized or aggregator platforms, as introduced in Section 2.1.4, can often only hardly be used for recruiting participants for network measurements. On these platforms, the workers might already be pre-filtered and limited to specialized or local groups, or the platform might not support dedicated Crowdsourcing use-cases. Crowd providers are more suitable here, but these platforms differ in their terms of use. For example, MTurk restricts tasks, such as asking workers to download and install software or to register at other web pages, whereas platforms such as Microworkers do not impose such restrictions. These restrictions must hence also be considered when designing the measurement tools, e.g., by selection of an appropriate platform or by designing a web-based tool.

Privacy and security constraints of the workers always must be considered, independent of the regulations of the platform providers. Running a software tool from an unknown employer imposes always a certain risk on a worker. Therefore, users may try to use sandbox environments to run the software or use fake identities to participate in tests that require registration. This, in turn, can result in biased measurement results. One possible solution is the use of web technologies like JavaScript that are widely supported and by design have no access to local data on the device.

SWOT Analysis of CNM

To further examine CNM, we continue with a SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis of CNM, which is graphically summarized in Figure 5.1. The large number of potential measurement points and the relatively low costs for conducting the measurement are some of the strengths of CNM. The most important benefit of CNM is the direct access to end user devices, which is also the main weakness of the approach. When using end user devices as measurement probes, the measurement software must be robust in face of different hardware and software environments, the technical capabilities of the probes might be limited in some cases, and the experiments are harder to control.

Nevertheless, CNM opens new opportunities for conducting measurements in realistic end-user environments, particularly large-scale user studies. Concerning threats, CNM results might be biased by unknown influence factors, e.g., due to limitations of end user devices or malicious workers. The success of a CNM



- Support for large-scale measurements
- Low measurement cost and fast execution
- Access to end user devices

- Diverse test environments
- Lack of control and coordination
- Limitations of measurement probes and privacy constraints

- Evaluation of realistic scenarios
- Analysis of end user behavior
- Large-scale studies at low cost

- Unknown factors influencing the results from measurement probes
- Acceptance of a measurement campaign by the workers is not predictable
- Some experiment setups not suitable for crowdsourcing tests

Figure 5.1: SWOT analysis of CNM

is difficult to predict, since a successful experiment depends no longer only on technical factors but also on the willingness of the workers to participate.

5.1.3 Comparison of Existing and Crowdsourcing-Based Network Measurement Techniques

Depending on the research question addressed, different numbers, locations, and technical equipment for the measurement points are required. For some tests, e.g., real user feedback must be included. In the following, we discuss in detail possible parameters for a network measurement setup and to what extent the test requirements could be fulfilled with different network measurement techniques. To illustrate the discussion, we use concrete examples, although the addressed parameters are applicable to a wide range of measurements. Thereafter, we directly compare the different measurement approaches and summarize their capabilities in Table 5.1.

Parameters of Network Measurements

The *granularity of network measurement data* is one of the parameters that must be considered when designing a measurement. The granularity of data can vary from packet traces to aggregated flows on a single client, e.g., to monitor application behavior at the end-user device [30], to traffic statistics of a backbone link for dimensioning wide-area networks [127, 128]. Although ISP traces allow the broadest spectrum of measurement granularity, the large amount of data requires the analysis of aggregated measures. In distributed testbeds, packet and flow-level data are available, but it is not possible to measure backbone link utilization. The data granularity from voluntary measurements and CNM is limited by the constraints of the end-user device as the OS might restrict collecting certain information. Concerning the examples mentioned at the beginning, distributed testbeds and CNM would be suitable for monitoring application behavior on a client device; however, ISP traces would be more appropriate for dimensioning decisions.

In addition to the granularity of the measurement data, experimenters also must adapt the *layers of the network stack* at which the data is collected according to the experiment. With custom ISP measurement solutions, it is possible to measure at any layer of the network stack as the hardware can be fully controlled. However, it is not possible to analyze application-layer data encrypted at the end-hosts. In a testbed, some network layers might not be accessible due to restrictions of shared testbeds. In voluntary participation and CNM, the measurement tools run on standard OSs with their security limitations and technical constraints. This often limits access to the network stack significantly.

Considering the analysis of a cloud office service such as Google Docs, ISP measurements would enable the experimenter to gather information on the lower layers of the network stack. For example, this would allow deriving traffic patterns of this service. However, cloud services such as Google Docs are usually secured by SSL/TLS connections; therefore, gathering information about the user's interactions with the service's web interface would not be possible using ISP traces and deep packet inspection as the payload is encrypted. The interactions can only be captured directly on the end-device, e.g., via browser plugins, after decrypting the packet payload. Consequently, voluntary participation or CNM could be used here.

Another measurement parameter is the *scale of a measurement*. The scale can be defined, e.g., by the number of measurement points, their geographical distance, or their distance in terms of inter-AS hops, i.e., number of hops within an autonomous system (AS). ISP-based measurements are limited by the number of nodes available to the ISP and the operations area of the ISP. Moreover, ISP nodes are naturally located in the same AS or in densely connected ASs. Testbeds in contrast, can scale from a few to several hundreds nodes, which are located at a single local site or are globally distributed. Global testbeds generally include nodes from multiple ASs, but they are largely located in research facilities and are therefore likely to be connected to dedicated broadband access networks. Voluntary participation often accesses a huge number of end-user nodes on a global scale, which are located in different ASs. However, the scale of the measurement, in terms of participating nodes, geographical distances and inter-AS distances, is not controllable. CNM is comparable to voluntary participation, but provides means of adjusting the scale of the measurement by hiring a dedicated number of participants from selected geographical locations. An example for a small-scale setup could be a local WiFi installation for interference tests. Here, the probes must be located close together, which usually requires a dedicated testbed. A possible application for a large-scale measurement setup is, e.g., the analysis of content distribution networks [129]. In this case, worldwide distributed measurement probes from different ASs are required, which can be achieved with voluntary participation, CNM, or a global testbed.

In addition to a sufficiently sized measurement setup, a certain *diversity of the measurement points* is also needed to achieve results representative for a larger number of real network users. ISP and testbed-based measurements are conducted using servers or dedicated measurement hardware, which are not common end-user devices. The same applies to the type of network access of nodes. CNM and voluntary participation offer a diverse set of hardware devices, such as end-user PCs, tablets, or smartphones. For both approaches, the type of participating measurement nodes can be influenced by providing specialized measurement software, e.g., only for Linux or iOS. In the future, crowdsourcing platforms might additionally allow hiring only users with given device specifications. Therefore, CNM and voluntary participation enable the evaluation of device-specific influence factors, e.g., on the traffic patterns of web applications, whereas ISP and testbed-based measurements allow for conducting reference measurements with comparable hardware and software configurations.

All network experiments require *control of the test environment* to a certain extent. This includes software tools, scheduling of measurements, and adaption of experimental parameters. Professional monitoring solutions are available in production environments of ISPs, but it is difficult to install experimental software tools or to influence the network significantly for test purposes. Testbeds in contrast offer a highly configurable and occasionally fully controllable environment, in which arbitrary software can be installed and be manipulated according to researchers' needs. Voluntary participants or crowdsourcing workers can be asked to install experimental software tools, but remote control of the tools is generally difficult to achieve. In both cases, the network parameters can hardly be influenced.

The *time scale* of a measurement is another parameter in the design of network measurement. It can vary from a single snapshot to a long term measurement, periodic measurements observing changes over time. Single-snapshot measurements are possible using any measurement technique. Long term and repetitive measurements, however, are more difficult to conduct in voluntary participation and CNM as the measurement probes must remain active over a longer period. Repetitive measurements using the same nodes multiple times are also difficult to achieve with voluntary participation and CNM, since the availability of the nodes is not guaranteed. In CNM, this issue is reduced as a group of workers can be hired again to redo the test. Using hired workers also helps to enforce time constraints, which are usually more difficult to guarantee in voluntary participation approaches.

In addition to technical parameters, *human factors* have become increasingly relevant in network research. On the one hand, end-users generate traffic patterns through their interactions, which can affect the infrastructure to a high degree. On the other hand, the QoE becomes an important factor in measuring the satisfaction of customers. ISP traces already include a realistic traffic pattern from end-users, but it is not possible to trigger specific user interactions, e.g., flash-crowds. Testbeds usually do not produce real end-user traffic, but when using synthetic generators, predefined traffic patterns can be emulated. Using voluntary participation can help to collect real end-user traffic, but it is difficult to trigger large-scale behaviors involving multiple users. CNM also offers access to realistic traffic, and even the triggering of flash crowds is possible. Furthermore, voluntary participation and CNM enable direct collection of actual user feedback.

Finally, the *costs* for conducting an experiment must be considered. The costs for using ISP traces or a testbed vary based on the point of view. Both measurement techniques require significant investments for the hardware and software necessary for the measurement and test infrastructure. However, after this infrastructure is set up, the costs for conducting measurements are relatively small. Voluntary participation and CNM require less initial investment cost, since only a reporting system is required to which the measurement results from the probes are sent. However, in CNM, every measurement introduces additional costs for the workers' salary and the commission for the crowdsourcing platform.

Table 5.1 summarizes the different parameters to consider when setting up network measurements and which realizations of those parameters are possible with the presented network measurement techniques. This overview can be used to select an appropriate measurement technique based on the measurement's requirements.

	ISP Measurements	Distributed Testbeds	Voluntary Participation	CNM
Granularity of measurement data	Packet level to backbone aggregate	Packet and flow level	Limited by constraints of end-user device	
Measurement layer	Any, except application layer	Partly network layer, no application layer	Application layer, additional information about real end-user devices	
Measurement scale	Limited to owned nodes; limited ASs; geographically close; fixed scale	Global scale; multiple ASs; fixed size	Global scale; multiple ASs; unpredictable scale	Global scale; multiple ASs; scale and location can be controlled by hiring participants
Diversity of measurement points	Dedicated measurement/server hardware		Realistic end-user devices; often devices in research networks	Realistic end-user devices
Controllability of the measurement	Professional monitoring tools available; experimental software cannot be deployed in production environment	Highly configurable; experimental software can be deployed	Experimental software can be deployed; remote control of software difficult to achieve	
Time scale	Snapshot and repetitive; short to long term		Snapshot; short term; repetition with the same nodes difficult to achieve	Snapshot; short term; repetition with the same nodes difficult to achieve, but possibility to hire the same people again
End-user interactions	Realistic interactions recorded in the traces; specific interactions cannot be triggered	Mostly synthetic traffic	Interactions can be measured and to a certain extent can be triggered	Interactions can be measured and triggered
Costs	Significant investment costs, thereafter free to use		Only expenses for reporting infrastructure	Expenses for reporting hardware; worker payments

Table 5.1: Network measurement parameters and their feasibility with different network measurement techniques.

Comparison of Network Measurement Techniques

Measurements performed by *ISPs* and in *distributed testbeds* are primarily taken using dedicated and specialized hardware. This enables deploying specialized measurement tools that gather information on the network layer. However, restrictions are imposed either by test isolation considerations in testbeds or by security constraints in production environments. Direct control over measurement probes enables long term and repetitive measurements. However, the specialized hardware and the dedicated testbeds impose biases on the measurements, which do not reflect end-user conditions. Both measurement techniques also fall short in providing direct end-user feedback or information about realistic end-user devices.

Voluntary participation and *CNM* measurement probes are intended to be real end-user devices. Consequently, the availability of individual measurement nodes varies significantly as users may go offline or only participate in a single test. Moreover, the duration the spontaneous participants contribute to a measurement cannot be predicted. However, both measurement techniques offer a rather realistic view of currently used end-user hardware and software and of end-user network connections.

The main difference between voluntary participation and CNM is the motivation of the users: Voluntary participation is based on altruism or non-monetary incentives, while crowdsourcing workers are mainly profit oriented. This results in different challenges when designing measurement tools. Software for voluntary tests must consider the incentives but does not necessarily require security features to identify cheaters. Moreover, voluntary tests require a certain amount of public relations management to build up and maintain a user base.

CNM, using monetary incentives in contrast, can be deployed rather quickly as the required number of participants can be directly recruited. However, the software must implement features to avoid cheating and fraud. CNM can also be used to kick-start voluntary participation approaches by recruiting the initial users.

5.1.4 Exemplary Use-Cases for Crowdsourcing-Based Network Measurements

After discussing the applicability of CNM to certain aspects of network measurements, we now illustrate the applicability of CNM on a exemplary use cases.

Realistic End-User Probes

Network measurements are mostly performed using dedicated testbeds. However, they only allow a biased view as the hardware and the broadband connections are not representative of real end-user systems. For instance, a significant share of PlanetLab nodes is located within a "Global Research and Education Network" [130] and the available bandwidth of the nodes and real end-users show large differences. To get an impression of this difference, we conducted a measurement of the access bandwidth of 500 Microworkers.com users in July 2013. In this measurement, we ask them to perform a commercial speed test [131] and hand in the link to the evaluation page showing their individual results. In April 2014, we conducted a similar measurement using the command line tool of the same measurement provider [132] on 163 PlanetLab nodes, resulting in an average download bandwidth of 174.8 Mbps (Std.: 216.2 Mbps). The results of both measurements are depicted in Figure 5.2.

Even if we only consider users from America and Europe, where the access speed is significantly higher than in Asia, the average download bandwidth of 17.21 Mbps (Std.: 24.09 Mbps) remains low compared to PlanetLab nodes. This confirms that network measurements performed on PlanetLab nodes may not be representative for real end-user devices, as already pointed out before by Spring et al. [116]. Therefore, additional reference measurements using at least a few end-users probes might be advisable for future measurement studies. The measurements also show that CNM might suffer from biases due to geographical location of the workers. However, the geographical location of a worker can be monitored and used during the evaluation phase to normalize the results.

For a second example, we conduct a study [21] about global expansion of the

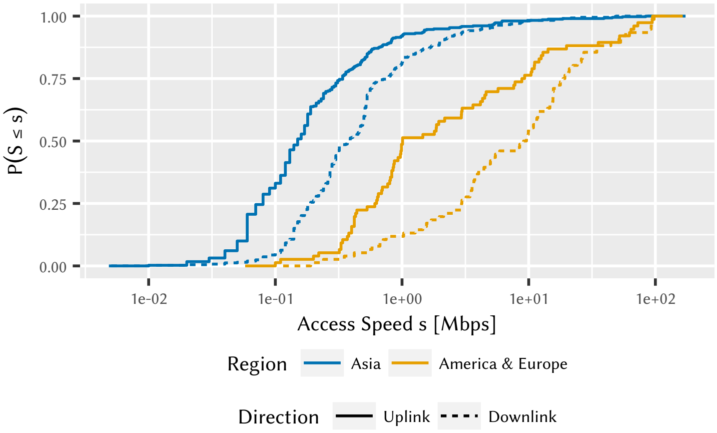


Figure 5.2: Measurement of access speeds of end-users obtained via CNM.

YouTube CDN by resolving physical server IP-addresses for clients in different locations. Both Crowdsourcing users recruited from Microworkers.com and PlanetLab nodes are used as measurement probes. During our measurement, most of the available PlanetLab nodes were located in the US and Western Europe. This is simply caused by the geographical distribution of the participating facilities. The CNM measurement was available to all workers on Microworkers and due the platforms demographic structure most participants were based in Asia-Pacific and Eastern Europe. The results from the measurement show that the capability of PlanetLab to measure a global CDN is rather low, since 80% of the requests are directed to US servers. In contrast only 44% of the requests from the Crowdsourcing users are directed to servers located in the USA.

In a second step, we analyse the number of different ASs the YouTube servers are located in and the number of YouTube servers per AS as observed by the PlanetLab nodes and the crowdsourcing users. Figure 5.3 shows the probability that a server belongs to AS with rank k. To improve the readability of the figure

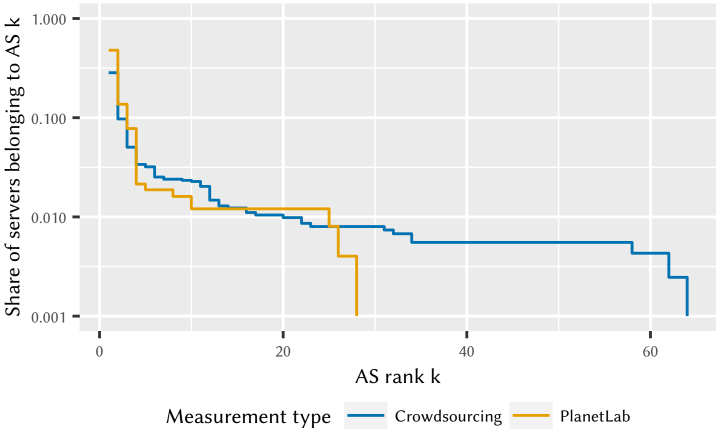


Figure 5.3: AS Distribution of YouTube servers as observed by CNM and PlanetLab.

the ASs were orderd according to their rank k, which is based on the number of YouTube servers within the AS. The measurements show that the PlanetLab nodes observed fewer than 30 ASs, whereas the Crowdsourcing nodes were able to detect more than 60. This indicates that CNM enables a more diverse view on the network than PlanetLab.

QoE and Application-Layer Measurements

One of the major drawbacks of testbed and ISP-based network measurement techniques is that no end-user feedback can be collected. In contrast, CNM and voluntary approaches can be used to conduct large-scale QoE measurements of real applications. To achieve this, measurement tools can be deployed on the participant's device to monitor network parameters, the application behavior, and collect real-time user feedback. With specialized test applications emulating a given application behavior QoE-influencing factors can be pinpointed even more easily. In addition to the information retrieved during the measurement, details about the workers are commonly available via the Crowdsourcing platform, e.g., the worker's country of origin. This allows identifying additional influence factors, like cultural biases that influence the subjective ratings [61, 133], or reducing the number of questions to the users for collecting relevant data.

In contrast to voluntary participation, the costs for CNM are higher, but CNM enables a faster completion of the test. In [21], we describe a QoE experiment with both voluntary users from social networks and paid crowdsourcing users. Whereas it took approximately 26 days to acquire approximately 100 voluntary testers, the same task was completed within 36 hours using a commercial crowdsourcing platform at a total cost of $16. More details on how to conduct subjective studies in a Crowdsourcing environment are given later in Section 5.2.

CNM and voluntary participation also offer easy means to gather information on the application layer. Whereas ISP traces only allow indirect information gathering of application information by analyzing packet and flow content, CNM and voluntary participation allow direct access to certain application information directly on the end host. The same is also possible using test beds; however, no real end-user interactions are available. One example for gathering application information using CNM and voluntary participation is given in [23]. In this study we implement a Dropbox application using the service's official API to gather meta information about the participants' account, e.g., the available and used Dropbox space. This enables us to collect objective information without any possible errors introduced, e.g., by erroneously survey results.

Exemplary results from this study are shown in Figure 5.4, depicting the CDF of the used Dropbox space of the participants. We observe significant differences between the volunteers and the Crowdsourcing participants. This indicates that one or both of our test groups show biases, but from the measurement itself, it is not clear which of them is biased. However, the Crowdsourcing results are in good accordance with other measurements [134], which suggests that the crowd-based measurements are more representative than the results obtained from volunteers.

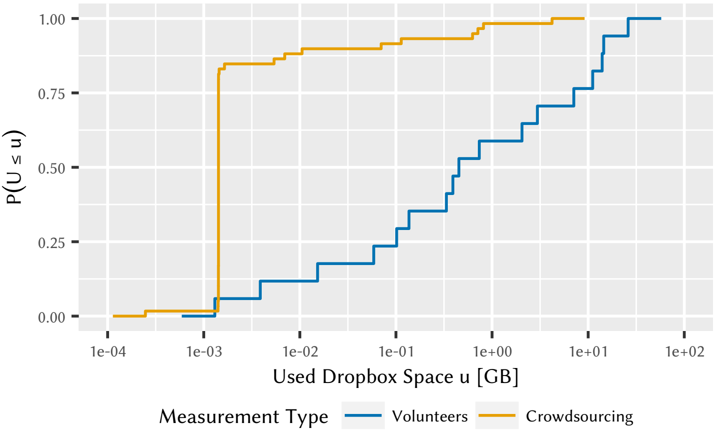


Figure 5.4: Usage of Dropbox space measured with crowdsourcing and with voluntary participation

5.1.5 Best Practices for Crowdsourcing-Based Network Measurements

Most of the crowdsourcing workers lack education in computer science or any experience in network measurements. Therefore, an *easy-to-use measurement software* is needed as executing complicated shell commands is too complex and error-prone. In contrast, the required interaction between the worker and the software should be kept at a minimum, with the user interface of the software being as simple as possible. This implies that also the results are automatically collected on a server to avoid any need for workers to address the result delivery.

Moreover, the software and the test design must *consider limitations of the target hosts*. Workers perform their tasks on a variety of devices and OSs. They might lack administrative privileges on their computers, e.g., in Internet cafés. Under these conditions, installing software and executing certain shell or scripting languages might not be possible. Therefore, we suggest using JavaScript or Java applets, since they permit running the measurements directly within the browser of the worker. An analysis of 558 workers from Microworkers showed that 97 % have JavaScript enabled and 53 % have a working Java Runtime Environment. However, the security mechanisms of the browsers prevent the execution of shell commands. Good reasons exist for this behavior, but it limits the application capability for network measurements; thus, workarounds must be developed. The size of the measurement software should be kept to a minimum as the bandwidth of the end-user device is usually limited. Hence, a trade-off exists between complexity and prerequisites on the one hand, and the number of successfully completed tasks on the other hand. This trade-off must be considered carefully during the design of the measurement campaign.

Moreover, there should be special focus on *choosing the right crowd-provider*. There are many options for recruiting participants including online social networks, online panels and a multitude of paid crowdsourcing platforms. All providers differ — occasionally significantly — in the supported types of tasks, demographics of their users as shown in Section 3.1, and their features for employers and workers [135]. In particular, the platform access, the diversity of participants, the costs per task and for qualification tests, payment features, the performance to acquire testers, and the integration of the measurement software into the platform must be considered. A comprehensive overview of all available platforms is not possible due to the large number. We provide a summary of features from two exemplary commercial crowdsourcing platforms and one possible source of voluntary participants in Table 5.2. Note that platform implementations and their features typically evolve over time. The information provided in the table reflects the status at the end of February 2016.

Depending on the specific requirements of the intended measurement, a careful selection of an appropriate crowd provider is needed. Demographical biases, e.g., can be avoided by filtering the participating workers or by selecting participants from multiple platforms, e.g., Facebook and a commercial platform, to ensure the required diversity. Limitations in terms of supported tasks are more dif-

	MTurk	Microworkers	Facebook
Platform access	Only US residents are allowed to create campaigns	Support of international employers	Support of international employers
Diversity of participants	Primarily US and Indian workers	International workers with a large portion from Asia	Primarily friends or acquaintances
Costs per task	One cent to a few dollars (depending on the task length)	Ten cents to a few dollars (depending on the task length)	free
Variable payment features	Bank transfer, Amazon.com gift cards	Micropayment services, wire card, credit card	Not applicable
Costs for qualification tests	Free	Ten cents to a few dollars (depending on the task length)	Free
Effort to acquire a large number of testers	None	None	Test must to be designed in a joyful manner to attract participants and to go viral
Time to acquire a few hundred of testes	A few hours to a few days	A few hours to a few days	A few days to a few weeks
Support of specialized participant groups	Worker groups can be selected by qualifications e.g., obtained by qualification test or overall performance, or by given attributes e.g., country	Worker groups can be selected by overall performance, special attributes, e.g., country, and deliberately formed by selecting individual workers	No direct support of grouping participants
Integration of measurement software into the platform	Forms are directly supported, more complex tasks must be implemented on an own server and embedded in an iFrame	Forms are directly supported, more complex tasks must be implemented on an own server and can be embedded in an iFrame	Tasks must be implemented on an own server and can be embedded using an iFrame

Table 5.2: Comparison of two commercial crowdsourcing platforms, Amazon Mechanical Turk and Microworkers, and the Facebook social network

ficult to address, and different implementations might be required. While some crowd providers such as Microworkers.com allow employers to pay for downloading and installing software, this is not possible on MTurk. Browser-based solutions using JavaScript or Java applets can still be deployed here.

Independent of the crowdsourcing provider, most crowdsourcing tasks are performed via a web interface, e.g., by showing images and providing an input field where tags can be added. However, CNM often impose requirements on workers' devices or network connections as specialized measurement software must be executed on the devices. Therefore, not all workers are able to complete the task if the device does not fulfill the experiment requirements. *Automated checks of the measurement prerequisites* at the beginning of the task can help here to minimize the time a worker spends on a task he cannot complete.

Consider a measurement setup containing a Java applet. The experimenter should automatically check whether Java is installed at the very beginning of the task. If Java is not available or enabled on a worker's device, detailed information can be provided why the task is not available for the specific worker. Checking the measurement prerequisites automatically also yields insights about possible issues with the task design, e.g., why most workers do not complete the measurement. Furthermore, detailed information about the end-user device can also be used to create personalized measurement settings for each worker, e.g., workers with more powerful devices can perform more repetitions than workers can with mobile devices.

Even if the measurement software is operating correctly, the workers might have problems in understanding the task. Thus, it is also important to *describe tasks in a clear manner and simple words.* To let a large number of workers complete the task successfully, its description must be easy to understand. Step-by-step instructions and screen-shots help workers to complete the task in a short amount of time. Technical and scientific terms should be avoided. Considering the large number of non-native English-speaking workers on international crowdsourcing platforms, a multilingual task description can also increase the completion rate of tasks.

Although the task description is detailed and well structured, some workers might face problems with the given task. Thus, it is necessary to *provide support for worker feedback and questions.* Feedback forms, forums, or email communication can be used for this purpose. Simple forms are recommended for optional feedback on the task, since they neither impose any additional effort on the worker nor reveal any additional private information such as email addresses. However, feedback forms only provide a one-way communication channel from the worker to the employer. This can be a significant disadvantage, e.g., if the workers faced issues during the task execution that cannot be reproduced. Email communication or forums can help here, since they enable more-interactive communication. However, according to our experience, forums should be preferred. In most cases, a majority of the workers face the same issues or have the same questions. Therefore, a forum thread can help to answer multiple questions at once or provide possible solutions to a large number of people with a single post. Furthermore, no private information, i.e., the email address, of the worker is revealed.

Using worker feedback, the employer can support the workers, improve the task description, or modify the task design if required. In multi-step tasks, feedback should be possible at every stage to let users who cannot complete the task ask questions. The employer must monitor existing communication channels of the workers often used to publicly discuss about unclear and erroneous tasks, e.g., public forums or the Facebook page of the platform operator. During one of our campaigns, a worker stated on the official Facebook page of the platform operator that his virus scanner detected malware in our software. The problem arose as the software tried to access the Internet for the measurements. A short post explaining the measurement details solved the problem and let the other workers continue our task.

Invalid measurement results are not only caused by misunderstandings or errors in the task design but also can be caused by cheating workers who try to receive the payment without performing the tasks properly. Therefore, *cheat detection and avoidance* techniques must be applied. The results from cheating workers can highly affect the results of measurements [26] and impose additional costs as shown in Section 4.2. A defensive task design, i.e., making it easier to complete the task in a meaningful manner than to find a means to cheat, can be applied to measurements in which no user interaction is required. If user interaction is required, e.g., the worker must access certain web pages or videos, such interactions can be monitored [20, 85, 136] or additional validation questions [26] about the content of the visited pages or videos can be added to verify correct task completion by the worker.

5.2 Crowdsourcing Based Quality of Experience Tests

We showed in the previous section that crowdsourcing is suitable for acquiring distributed and realistic network probes. Besides the device access, the main benefit of crowdsourcing based network measurements is the possibility to collect subjective ratings on service quality. However, subjective testing is an integral part of not only in network research but also the research on multimedia technology and algorithms, as any new concept needs to be validated with respect to the suitability for the potential users. Besides usability, acceptability and performance, the users' overall Quality of Experience (QoE) in the context of multimedia applications is often a focus of subjective tests. These tests, however, are expensive from both an organisational and a financial perspective: test subjects need to be recruited and test sessions need to be organised, often with constraints on the number of test subjects that can participate simultaneously in the laboratory, leading to time consuming test campaigns and a lack of flexibility. Furthermore, due to the fixed location of the laboratory, the subjects may not be a representative sample of the complete population in a statistical sense. Additionally, test subjects often need to be reimbursed on a competitive wage level in order to get a sufficient number of test subjects. Thus subjective testing can often strain the available resources, resulting in either a compromise in the number of considered test cases or avoiding the subjective testing altogether.

QoE crowdtesting provides an alternative to the traditional subjective testing, aiming at reducing the resources necessary for conducting subjective testing by utilising crowdsourcing. Even though the use of the Internet as a virtual laboratory leads to limitations on the stimuli and scenarios that can be tested, the ever increasing bandwidth and capabilities of the connected devices allow for a wide range of areas in which QoE crowdtesting can be used. QoE crowdtesting, however, is not just a straight forward implementation of existing subjective testing methodologies in an Internet-based environment. Owing to the fundamental differences between the traditional and virtual laboratory, extra considerations need to be taken in order to gain reliable results.

In this section, we therefore provide a collection of best practices for QoE crowdtesting by addressing on the one hand the key issues that need to be considered if a subjective test should be replaced by QoE crowdtesting, and, on the other hand, how these issues can be addressed best in the design and implementation of the desired QoE crowdtesting campaign. We have chosen the QoE assessment of videos as an example to illustrate the proposed best practices, but they can also be applied to the QoE assessment of other stimuli. Some of the best practices proposed in this section show some overlap with the best practices for CNM.

However in the case of CNM crowdsourcing workers are mainly recruited to gain access to their end-user devices for obtaining objective technical measurements. Thus, the main challenge of CNM is to design an appropriate (highly-automated) measurement tool for the crowdsourcing environment. In contrast QoE crowdtesting focuses on collecting subjective ratings from the crowdsourcing workers themselves. Here, the specific workers play a much more important role and consequently the design of the task itself including, e.g., the presentation of the instructions, incentives, and interface design, are even more important than for CNM. Additionally, the aggregation of the subjective results and the identification of erroneous submissions is more challenging than for the technical measurement results from CNM.

The remainder of this section is structured as follows. The key issues of QoE crowdtesting are summarized in Section 5.2.2 addressing limitations, reliability, incentives and task design, context monitoring, and hidden influence factors. Technical challenges and best practices for the implementation of QoE crowdtesting are analysed in Section 5.2.3. The statistical analysis of the obtained user ratings from QoE crowdtesting is shown in Section 5.2.4, where we show the need and mechanisms for filtering out unreliable user ratings. Based on the preceding sections, we then summarize the proposed best practices in Section 5.2.5. The content used in the remainder of this section was previously published in [3].

5.2.1 Background on Quality of Experience and Video Quality Assessment

One possible definition of QoE in the context of multimedia systems and applications is provided in [137] as “the degree of delight or annoyance of the user of an application or service. It results from the fulfilment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the users personality and current state”. Following this definition, QoE is influenced by a variety of factors [137, 138] that can be divided into four different categories, each representing a different level in multimedia systems and applications: *context, user, system*, and *content* level.

The *context level* considers aspects like the environment in which the user is consuming the service, the user’s social and cultural background or the purpose of using the service, e.g, recreation or information retrieval. The *user level* includes psychological factors like expectations of the user, memory and recency effects or the usage history of the application. The technical influence factors are abstracted on the *system level.* They cover influences of the transmission network, the devices and screens, but also of the implementation of the application itself like video buffering strategies. Lastly, the *content level* addresses characteristics of the content, e.g., for video, the video codec, format, resolution, but also duration, content of the video, type of video and its motion patterns.

The aim of QoE crowdtesting is to move the QoE assessment from a standardized lab environment into the Internet, where the crowdsourcing platforms act as an extra layer between test manager and test subject, handling the recruiting and payment of the test participants. The subjective testing is therefore using subjects from a global worker pool, usually with a web-based application, that can be accessed via common web browsers.

Video QoE assessment is done for a range of different application areas: from the visual quality evaluation of video coding technologies and processing algorithms to the influence of network delays and packet loss on the video quality. The QoE of video is usually determined in a well-defined testing environment with subjective methodologies, as described in standards like [139, 140]. However, in the context of QoE crowdtesting, we must distinguish between two categories of video QoE assessment: QoE evaluation of Internet-based video applications for instance YouTube and QoE assessment of video in general such as the evaluation of coding technologies. The difference between these two categories lies in the fact that the Internet-based video applications are by design optimized for the presentation in a web environment and can therefore be easily adapted to QoE crowdtesting. In contrast, applying crowdtesting to video QoE assessment in general necessitates the additional design of Internet-based applications for the presentation of the videos under test. Both categories will be discussed briefly in this section.

QoE crowdtesting of Internet-based video applications is relatively straightforward, as the main difference to the lab is the use of crowdsourcing platforms for test subject recruitment and reimbursement. Although some adaptations for interfacing with the crowdsourcing platforms may be necessary, the application itself needs not to be modified. One typical example of this category is examining the influence of stalling events in video streaming on the video QoE as discussed, e.g., for YouTube in [26]. Here, the test setup in the lab usually also consists of a web interface presenting the videos and collecting the subjects' scores. However, in order to avoid additional stalling caused by the test users' Internet connection, the videos have to be downloaded completely to the browser cache before playing. During the initial download of the videos, a personal data questionnaire may be completed by the participant including also consistency questions to check for reliability [26].

For general video QoE assessment, the adaptation of the lab tests to QoE crowdtesting is more cumbersome. Firstly, the testing methodology needs to be provided by an Internet application, instead of platform-dependent software. Secondly, the delivery of the videos under test must be implemented. Especially for testing methodologies, that use an uncompressed video for comparison, this requires dedicated applications. Alternatively, a video crowdtesting platform like *QualityCrowd* [53] can be used, that already takes these issues into consideration. In addition, it may also be necessary to adapt the goal of the test to the limitations of the crowdsourcing environment, e.g., videos with spatial or temporal resolution beyond the capabilities of consumer equipment need to be down-sampled.

Common to both categories is that instead of a sophisticated hardware and standardized test environment, the hardware and viewing environment will vary between the different workers. This *lack of control* can be tackled using the different strategies of *monitoring*, *adaptation* and *prevention* as will be discussed in detail in Section 5.2.2. In contrast to these environmental issues, however, common subjective testing methodologies for video quality assessment can be used. Using ITU-R BT.500 [139], e.g., both the discrete double stimulus and the continuous double stimulus method can be implemented easily in a corresponding web interface.

Studies from literature have shown that using crowdtesting for the QoE assessment of a wide range of video applications can deliver results similar to traditional testing in the lab environment: Keimel et al. [53, 141] have shown that crowdtesting delivers results within the acceptable inter-lab variation between different testing labs for standard conforming QoE assessment, Chen et al. [82, 142] discussed crowdtesting for audio-visual QoE of Internet-based applications, which was discussed more in detail by Wu et al. [143], and Hoßfeld et al. [26] applied crowdtesting to examine the influence of stalling events and initial delays [144] on the QoE in video streaming applications. For pairwise comparison QoE tests, Xu et al. [145]suggest an approach to decompose the pairwise comparison data onto random graphs, reducing the assessment tasks for each participant significantly and therefore making pairwise comparison more suitable for crowdtesting.

5.2.2 Key Issues in QoE Crowdtesting

We discussed earlier that crowdsourcing gives researchers new possibilities to conduct subjective user studies. For QoE assessment, similar to CNM, conceptual challenges arise by moving studies to the crowd. In the case of CNM, these challenges are mainly related to the design and the automation of the measurement software. For QoE crowdtesting the main focus lies on creating a test design that offers similar conditions as a controllable laboratory environment. Here, challenges arise especially as micro-tasks are typically short compared to long lab studies and the workers are heterogeneous with respect to their hardware and the environmental settings. Moreover, the reliability of the workers' ratings can vary significantly. In the following we have a closer look at these challenges and also the limitations of the QoE crowdtesting approach.

Limitations of QoE Crowdtesting

In principle, QoE crowdtesting could be used for the assessment of any stimuli and interactivity, using any type of subjective methodology. In reality, however, we are faced with several limitations on the possible scope of QoE crowdtesting.

The main technical factors limiting the scope of QoE assessment are bandwidth constraints and support of the workers' devices to present the required stimuli. The first factor requires to consider the support of coding standards by the workers devices, as it is often not feasible to provide the uncompressed stimuli to the workers due to excessive bandwidth demands. This is in contrast to the traditional lab setting, where the aim is to avoid any additional compression of the stimuli under test. But even with supported codecs, the size of compressed stimuli may be too large for the connection bandwidth of many workers, especially for HDTV or even UHDTV formats.

Secondly, the stimuli must be supported by the workers' devices. Although 2-D video and audio capabilities have become standard for most devices, 3-D video and audio capabilities or high dynamic range (HDR) displays cannot be readily assumed to be available. The support for other stimuli, e.g., haptic or olfactory stimuli, is nearly non-existent in common computer hardware as used by the workers and thus these stimuli are currently not suitable for QoE crowdtesting. Besides these technical factors, QoE assessment methodologies requirng the interaction between different workers, e.g., for interactive video conferencing, are possible, but challenging in their execution.

Summing up, QoE crowdtesting is feasible for 2-D video, image and audio QoE assessment tasks, where the usable formats depend on the bandwidth requirement. In particular, for video, HDTV formats, depending on the required bitrate, may not be suitable for QoE crowdtesting with today's Internet access speed.

Conceptual Challenges for QoE Crowdtesting

The migration to crowdsourcing invokes some *conceptual challenges* on how to assess QoE and how to design the user tests [146]. In laboratory studies user tests may take up to 90 minutes [147] which allow, e.g., to investigate memory effects [148]. In contrast, crowdsourcing tasks are typically rather short as discussed in Section 2. Therefore, tests designed for a lab environment need to be modified for crowdtesting and one of simplest ways to this is by partitioning the test into basic test cells [146]. As a consequence, a crowdsourced QoE test user may only see a subset of the test conditions which requires sophisticated statistical methods for outlier detection or quantifying reliability. Another issue with QoE crowdtesting is the lack of a test moderator, but the user is guided via the web interface through the tests. In particular, the training of subjects is different than in a traditional lab environment and is mostly conducted by means of qualification tests. Nevertheless, in case of any problems with understanding the test, uncertainty about rating scales, sloppy execution of the test, or fatigue of the test user, appropriate mechanisms or statistical methods have to be applied.

Unreliability of Users: Reasons and Task Design Solutions

There are several reasons why some user ratings are not reliable and need to be filtered out. *Technical errors* may occur due to errors in the web-based test application or due to incompatibilities of the test application with the worker's hard- and software including missing video codecs or insufficient screen resolution. As a consequence, the users observe different test conditions or additional artefacts occur, leading to test results which appear unreliable, but may be valid for the individual users' conditions. This requires an appropriate monitoring of the system. Another possible reason for unreliable user ratings are the *test instructions* which may not be clear or too complex to understand, and additionally *language problems* may also occur with international users. Furthermore, there may also be *cheating users* as discussed Chapter 4.

Incentives and Payment Schemes

Incentives play a key role in the successful use of crowdsourcing in general and QoE crowdtesting in particular. Incentive design addresses the development of mechanisms and presentation of the task according to the following two goals: On the one hand, incentive design aims to improve the willingness of subjects to participate beyond purely monetary interests, e.g. through gamification, and thus more users are completing the study in a shorter time. On the other hand, incentive design aims to improve the quality of the results generated by the subjects with incentive mechanisms that are complementary to reliability mechanisms [26] or data quality mechanisms [81].

While reliability mechanisms aim at filtering out unreliable users or unreliable results, data quality mechanisms try to estimate the quality of the workers or their submitted results in order to reject or block the low-performing workers. Different mechanisms for different domains have been proposed in literature: from image labelling [81] to natural language processing [149]. However, in the context of incentive design only a few insights and general conclusions are available. [150] shows that incentives encourage participants to make more accurate judgements when using crowdsourcing for screening a number of candidates applying for a job at a company and to conduct resume reviews. Positive incentives were represented by bonus payments: Each participant was initially told that each resume had already been rated by an expert and, if the participant's rating matched the expert's, the bonus was paid. In contrast, negative incentives were represented by telling the participant that their payment is reduced, if it differs from the expert's rating. A combination of positive and negative incentives was also applied. All incentive schemes in [150] increased the quality of work. Other payment schemes may depend on the actual performance of the worker, e.g., the user is allowed to "choose as many as they want" test sequences for QoE assessment, and then they are paid accordingly to the number of evaluated tests sequences.

Beyond payment schemes, other incentives address social aspects, entertainment and altruism [151]. Gamification or games with a purpose [152] is an approach to develop incentives for entertainment and fun, enabling human contributors to carry out computation tasks as a side effect of playing online games [149]. In the context of data or image labelling, different games are discussed in [91, 153]. However, there are no general guidelines how to design a game, as this is strongly task related. Nevertheless, the results for the gamification of tasks are very promising: Eichhoff et al. [153] show that 70 % of users played more than the first round necessary for the payment, i.e., the additional results are obtained for free by the employer. 80 % of the users return to a game, compared to only 23 % for a regular task and unreliable ratings in their task annotation game are reduced to 2.3 % instead of 13.5 %, compared to a non-gaming task. Innovative, creative tasks are less likely to be cheated on and also the time and cost is spent more efficiently. The quality of the results increased by 10 %. Thus, gamification has the potential to make crowdsourcing an even more powerful tool for QoE assessment.

Context Monitoring and Hidden Influence Factors

Due to the remoteness of the participants and the heterogeneity of the used soft- and hardware, it is necessary to monitor the users' environment in order to identify additional influence factors on the QoE assessment. Influence factors are defined as any characteristic of a user, system, or context that may have influence on the users' QoE [137]. Human influence factors are variant and invariant characteristics of a human user describing the demographic and socio-economic background, the physical and mental constitution, or the user's emotional state. System influence factors are related to the media coding, transmission, storage, rendering, and reproduction/display. The context influence factors describe characteristics of the users' current environment that may also influence the QoE. Due to the unknown context in which the QoE assessment is performed by the workers in QoE crowdtesting, these influence factors are not known beforehand, but hidden, yet still influence the users' QoE ratings.

In general, we have three options to cope with the unknown context and the resulting hidden influence factors. We can either monitor the appropriate context parameters, adapt the context or try to prevent the undesirable context itself in our test design. In the following, we highlight some examples for best practices.

Monitoring of the workers' environmental conditions and context is required, since the environment in which the workers evaluate the stimuli in QoE crowdtesting may impact the overall QoE and thus the application should be able to detect such factors. For visual stimuli, e.g., the general viewing conditions represented by the background illumination or the screen resolution itself can be influencing factors.

One option to adapt the conditions of the workers' environment is to provide them with simple test patterns that allow them to either calibrate their devices or enable the quantification of the deviation of a device's stimuli representation from the desired target. For visual stimuli, a basic test pattern similar to the test patterns used for calibration of the monitor contrast and illumination in a professional environment can be utilised to quantify the users' viewing conditions, e.g., by asking how many grey steps on a greyscale step-wedge are visible. Similarly, we can prevent an undesirable context from the technical perspective, e.g., for video QoE assessment, by pre-loading videos with included distortions in the remote browser, so that additional distortions introduced by the transmission do not affect the playback. This ensures that the influence of the users' context with respect to bandwidth is no longer an issue.

A hidden influence factor on the user level can be the *users' expectations*: those used to lower quality (e.g. low video resolution) will rate differently than those typically consuming higher quality (e.g. high video resolution). The expectation level may also be closely related to the country of the subject and users from different regions may have different expectations about the provided content quality. In general, we have two options to cope with expectations. We can either quantify the degree of expectations or we can reduce the expectations by instructing the test user accordingly. One option to quantify the expectations is to group users according to their expectations by asking them about their habits and typical use of a service, e.g., "How often do you watch Internet videos?" and "Do you watch low or high resolution in YouTube?", respectively, where the assumption is that subjects who do not use video streaming services often may be more tolerant to worse quality.

In the QoE rating task, a user may additionally be asked to rate on an extra expectation category scale that is better aligned with the actual user's expectations. The subjects then rate the perceived quality with, e.g., five levels of expectations: *(-2) Much worse than I expected. (-1) Worse than I expected. (0) Just as I expected. (1) Better than I expected. (2) Much better than I expected.* This rating scale is accompanied with a question regarding the perceived quality, e.g. *"Please indicate to which degree the overall quality of this video was in line or not in line with your expectations? The overall video quality was...".* Still, the quantification of expectations remains an open research topic.

Demographics and User Impairments may also have an impact on the QoE results and should therefore be statistically analysed. Different possibilities exist to acquire demographic information about a worker. One solution is to include a short survey in the task itself. However, it is not clear if the workers answer truthfully. This can be overcome with the use of consistency tests, but only a subset of data can be used in order to avoid overusing consistency questions. Another possibility is to extract information about the worker from the worker's social network profile, if it is known and the data is accessible. Finally, some crowdsourcing platforms also provide some demographic information about the workers on dedicated worker profile pages. Besides demographic information hidden influence factors on the QoE results may be caused by physical impairments of the subjects if they are crucial for the study. For visual stimuli, e.g., a test for colour blindness may be necessary to confirm normal colour vision if required in the test.

QoE crowdtesting are subjective tests conducted in a heterogeneous and therefore partly uncontrolled environment. Thus, monitoring of the *hard- and software environment* is required to analyse hidden influence factors on a system level. Due to bottlenecks at the end user devices in terms of CPU, memory, or network bandwidth, additional artefacts may arise and affect the user rating accordingly, e.g., the user's Internet access bandwidth may not be large enough to conduct a video quality test without stalling. However, those stalling events and the corresponding unintended freezing of the video will impact the QoE. To overcome the impact of the network delay due to Internet delivery of data, the test application and data may be completely downloaded before the actual user test starts. Even so, the resulting initial delays may also be too long and influence the user rating. In both cases, it is evident that monitoring on system level is required. As a possible solution, download speed and latency may also be measured before the actual test, and then only users are selected with suitable connection speed and latency.

5.2.3 Implementation and Design of QoE Crowdtesting Campaigns

While designing a QoE crowdtesting campaign, the well established recommendations for laboratory subjective assessments can be respected only to a certain extent. Time constraints and test complexity should be adjusted in regards to a web based or other crowdtesting scenarios and to the variety of testing subjects among the crowd. Moreover, QoE crowdtesting brings additional requirements on server capabilities, computing power, and resource management.

Therefore we discuss major challenges concerning the available resources, either on a server side or on a client side, and best practices regarding the implementation, as well as setting up the campaign. A sophisticated two-stage crowdtesting design is proposed and recommended.

Implementation

The general approach for QoE crowdtesting is the use of a dedicated *test server*. This allows for a specific and well controlled testing environment. The choice of a dedicated test server gives additional possibilities to perform application layer monitoring, which further enhances the overall efficiency and accuracy of the given QoE crowdtesting application. Moreover, supporting technologies, e.g., social networking or Crowdsourcing platforms' APIs can be easily implemented.

Depending on the actual requirements on computing power and network resources, third-party services, such as cloud computing services or content delivery networks (CDN) can be utilised. The choice of a third party cloud service strongly depends on the size and type of a targeted QoE crowdtesting panel of users. While the use of a dedicated testing server is beneficial with respect to the better control of the environment, the test designer should take into consideration that users in a QoE crowdtesting campaign are accessing the application from a whole variety of different places. Hence, the accessibility of the server is an important issue. CDNs are well adapted to this fact and allow for better accessibility of the whole application. A large number of participants in a survey can result in a significant amount of web traffic. This should also be taken into account when designing the recording system for the results, in particular with respect to the capability of handling a high number of queries in a short period of time. Apart from cloud services or CDNs, another suitable option is limiting the number of simultaneous users of the application. If insufficient computational power or network resources are available, it is beneficial to put the users in a waiting queue and inform them about the waiting time. However, a long waiting time could result in a decrease of successfully finished surveys. Also, the options for using cloud services or waiting queues are not mutually exclusive and, if needed, they can be combined.

The *client interface* may change significantly for the QoE crowdtesting, depending on the users' environment. This should be reflected in the basic application design and the implemented technologies should put minimal requirements on browser's capabilities. In particular, the designer should focus on widely available and adopted technologies, since users may access the application from locations such as Internet cafes, where they are unable to install additional software. According to [154], the technology mainly supported in web browsers is still JavaScript, while Java is on a decline, with only approximately 70% of the market share. Similarly, other technologies like Silverlight, QuickTime or Mediaplayer are representing less than 50-70% of the market share. Thus if the application requires support of not commonly used technologies, e.g., Java, it will cause substantial loss of workers who successfully finish the assigned task. Losing 30% of the subjects in a QoE crowdtesting survey can easily represent hundreds of people, causing a bias in the overall results and demographics of the crowd.

Support for all the required features should be properly tested before the beginning of the actual QoE assessment. These tests should include basic benchmarks, to ensure the browsers' ability to display the test correctly. For example, the video QoE crowdtesting application of Gardlo et al. [155] included tests for JavaScript and Flash support in the browser, Internet connection speed, screen resolution and the flaw-less playback of high definition videos. Note that an important point is to keep the benchmarking time as short as possible, for not interfering with the actual QoE crowdtesting, and thus distracting the users.

Setting up the Campaign

In the process of creating and setting up the new campaign, a fundamental question is the maximum *length of the test* tolerated by the worker. Despite the apparently rudimentary character of the question, the answer is rather complex, as the length of the test strongly correlates to several parameters, namely the overall *enjoyability* of the whole test, the complexity of the task, the user interface, the amount of reward, and the worker's ability to understand the task. Concrete guidelines for the optimal length of a task cannot be given but still are subject to ongoing research. Thus, only piloting phases can help to estimate a meaningful duration for a concrete task.

Similar to the design guidelines for CNM tasks, also QoE crowdtesting tasks should stick to a *very basic and transparent design*, with minimal requirements for user interactions, preferably many of them automatized. Regardless of the depth of the integration of automatized interactions, it is necessary to keep in contact with the user and inform him about the task's progress. It is beneficial to offer users the application *interface in their native language*, possibly even by only adding an automated translation plug-in to the page.

Workers should also be *properly rewarded* for the successfully finished task, with respect to the complexity and overall duration of the task. In [142], users get only paid depending on specific rules after successfully completing a task and achieving sufficiently high reliability scores. Better paid tasks will attract more users, but they will also be more critical to the application. Workers also tend to gather in virtual communities and share their experiences with certain employers, contributing to an employer's reputation and in extension the attractiveness of the tasks offered by this employer. Good payment and properly designed applications without any errors will be well received among the community, and this also helps to increase the overall efficiency of the QoE crowdtesting application.

Two-Stage QoE Crowdtesting Design

Current platforms do not implement sufficient quality assurance to ensure high data quality for QoE crowdtesting tasks. However, this can be overcome using an own test server with tailored quality assurance mechanisms.

Some platforms allow for the automatic monitoring of user reliability using gold standard data, which is common practice for many Crowdsourcing tasks. However, for QoE assessment gold standard data is not available in general. The ratings submitted by the workers are subjective and consequently there is no *correct* answer that can be used as ground truth. Yet, depending on the QoE assessment task, some secondary properties may be utilised as gold standard data, e.g., the number of noticed stalling events in a video, if the events were introduced artificially and it is known objectively that stalling occurred or not. Additionally, a dedicated test server allows the application designer to implement social networks' APIs, e.g., Facebook, and enables the employer to utilise the mutual advantages of each of these two distinctive crowdsourcing categories. We may, however, lose users without social network profile, but we also gain an important advantage by having demographic data available without any additional questionnaires. This reduces the overall testing time and the additional data can also be used in assessing the users' reliability.

The general recommendation for campaign settings is the ***two stage*** design. The ***first stage*** represents a very simple and easy to do task, which

- tests the reliability of the users,
- gathers a huge panel of users,
- gathers information about the users in the crowd,
- has a duration of less than a minute and low payment, and
- can perform context monitoring: Hardware or software, or perform user's training.

The intention of this stage is to create a *pseudo-reliable* group of users, who will be later invited to the actual crowdtesting task. An example of such an application is a simple screen quality test, where the user has to select visible pictures from a group of difficult-to-see or invisible images on a low quality screen. The task is easy to do, fast to finish and has low pay, so within a short period of time and with low costs it is possible to create a reliable panel of users. This stage significantly improves the overall efficiency of the whole campaign.

The actual QoE test is then conducted in the ***second stage***, only with invited reliable users from a previous campaign. However, it is important to test if the same hardware or software is being used as in the first stage, but also to test the users reliability, e.g., with content questions, demographic questions, or repetitive presentation of tested content. Note, that the use of hidden reference methods e.g. proposed in ITU-R BS.1116 [156] can be considered as consistency questions as suggested in the two-stage design. This stage also requires a higher reward for the workers. In the notion of ITU-R BS.116 [156], we also apply a pre-screening and a post-screening technique. The major argument for introducing the pre-screening, i.e. the first stage, is to reduce costs of the overall campaign and to get a pseudo-reliable crowd, while the post-screening in the second stage is required to ensure a reliable data set.

Although not necessary, it is reasonable that the task required of the workers in the first stage is related to the task in second stage, e.g., if the main task in second stage consists of a visual quality assessment, the first stage should also consist of a task including visual stimuli as in a screen quality test mentioned in the example above. Moreover, this can avoid any disappointment by the workers, resulting in decreased reputation of the employer, if the tasks in the two stages are very different. Also not every worker passing the first stage may be willing to participate in the second stage. In [157], e.g., up to 75 % of the workers passing the first stage declined to participate in the second stage.

5.2.4 Statistical Analysis of Test Results

The two-stage design and the general implementation guidelines recommended in the previous section address the key issues discussed in Section 5.2.2 and lead to an overall better reliability of the QoE crowdtesting results. But even tough more reliable, the results may still contain a certain amount of unreliable ratings and/or workers. Similar to any laboratory-based QoE assessment test, we therefore need to perform a statistical analysis of the results in order to identify these unreliable results. Unfortunately, methods based on user ratings commonly employed in subjective QoE assessment are not suitable in the context of QoE crowdtesting. We demonstrate the shortcomings of these methods on the example of two QoE crowdtesting studies on video streaming. Before evaluating the commonly used screening methods from literature, we briefly introduce the details of a QoE crowdsourcing study, followed by a demonstration of the severe concealed influence of unreliable ratings on QoE results. Appropriate metrics to report the reliability of the results of QoE crowdtesting campaigns are then suggested.

Existing QoE Crowdtesting Data for Further Evaluation

For the statistical analysis, the results from a subjective user study on video streaming, which we detail on in the following, are revisited with respect to reliability YouTube video streaming is considered, where impairments in the network are perceived as stalling of the video playout by the user. If the available network bandwidth is lower than the video bit rate, video transmission becomes too slow, gradually emptying the playback buffer until underrun occurs. If rebuffering happens, the user notices interrupted video playback, commonly referred to as stalling.

In the QoE crowdtesting campaigns [26, 144], different reliability mechanisms are implemented as discussed in Section 5.2.2. In particular, unreliable users are determined based on content questions, consistency questions, and gold standard data [26]. After watching a video, the users are asked to answer simple content questions about the video clip. For example, "Which sport was shown in the clip? A) Tennis. B) Soccer. C) Skiing." The users are asked about their origin country in the beginning and about their origin continent at the end of the test to check their consistency. As gold standard data, we include videos without any stalling and additionally ask participants: "Did you notice any stops to the video you just watched?". If a user then notices stops, we disregard his ratings. We also monitor the user's interactions as proposed in Section 4.1. In this particular case we record browser events in order to measure the focus time, which is the time interval during which the browser focus is on the website belonging to the user test. In order to increase the number of valid results from crowdsourcing, we display a warning message if the user did not watch more than 70 % of the video. The users can decide to watch the video again or to continue the test. When users became aware of this control mechanism, the percentage of completely watched videos doubled and almost three times more users could be considered reliable than without the system warning. In particular, we consider a user and all his ratings to be unreliable, if one of those questions is answered incorrectly or the video focus time is shorter than the video duration.

To have a realistic test scenario, the video experience in the test should mimic a visit of the real YouTube website. To this end, an instance of the YouTube Chromeless Player is embedded into dynamically generated web pages. With JavaScript commands the video stream can be paused, a feature we use to simulate stalling. In addition, the JavaScript API allows monitoring the player and the buffer status, i.e. to monitor stalling on application layer, by checking the current state of the player. In order to avoid additional stalling caused by the test users' Internet connection, the videos had to be downloaded completely to the browser cache before playing. This enables us to specify fixed unique stalling patterns which were evaluated by several users.

During the initial download of the videos, a personal data questionnaire is completed by the participant which also includes consistency questions mentioned above. The user then sequentially views six different YouTube video clips with a predefined stalling pattern. Typical YouTube videos of various content

Table 5.3: Details on the QoE crowdtesting data sets.

	YouTube
Impairment	Stalling: video interruptions and waiting times
Rating scale	Ordinal 5-point ACR scale
Test method	Single stimulus
ITU Rec.	ITU-T P.910
#Videos	21 typical YouTube videos
Video duration	30 s
#Subjects	722
Reliability	Gold standard, consistency and content questions, video focus time

classes like news, sports, music clips, cartoons, etc. are used in the tests. Thereby, the video clips have different resolutions, motion patterns and video codec settings, but a fixed length of 30 s. After the streaming of the video, the user is asked to give his current personal satisfaction rating during the video streaming on an ordinal 5-point absolute category rating (ACR) scale. More details on the test setup can be found in [26, 138].

Influence of Unreliable User Ratings on QoE Results

For the YouTube tests, the unreliable user ratings are determined based on content questions, consistency questions, gold data, and video focus time as described in the previous section. The results for the QoE crowdtesting campaign are depicted in Figure 5.5 as a CDF of the unreliable user ratings as well as the corresponding 95 % confidence interval.

The unreliable user ratings F can be approximated by a discrete uniform distribution U with values from 1 to 5. The average user rating $\overline{F}$ and the expecta-

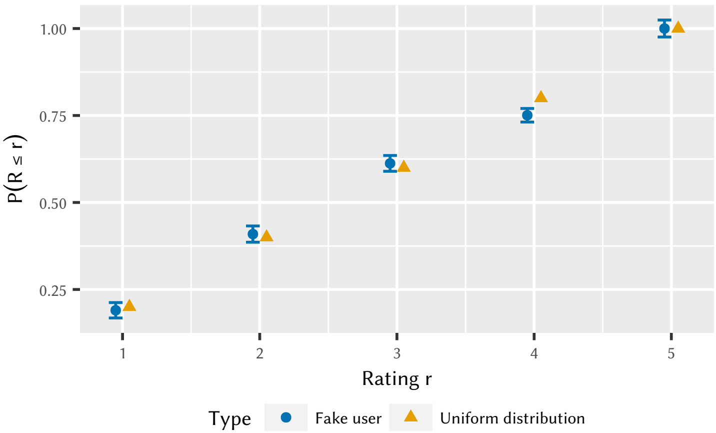


Figure 5.5: Unreliable user ratings for YouTube crowdsourcing tests identified by means of reliability mechanisms as described in Section 5.2.4.

tion value $\overline{U}$ are 3.04 and 3.00, while the standard deviations are $\sigma_F = 1.45$ and $\sigma_U = 1.58$, respectively. A Pearson's χ^2 test is performed with the null hypothesis that the unreliable ratings are uniformly distributed. At the 5 % significance level, the null hypothesis cannot be rejected with a p-value of 0.39 to observe the given statistic with probability p.

Figure 5.6 depicts the influence of unreliable user ratings on QoE results. In particular, the Mean Opinion Scores (MOS) and the corresponding 95 % confidence intervals for YouTube experiments are plotted depending on the test conditions, representing the number of stalling events during the YouTube video playout in that case. The reliable user ratings from the YouTube experiments (237 in total) are therefore considered in presence of a ratio of α unreliable ratings in relation to the overall number of ratings. According to Figure 5.5, the unreliable user ratings are drawn from a uniform distribution between 1 and 5.

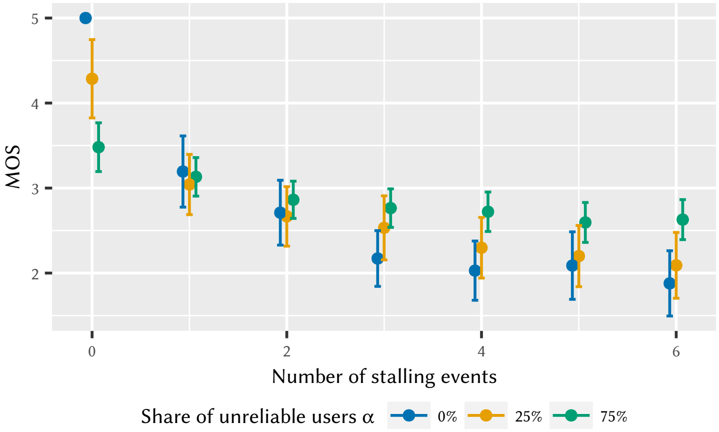


Figure 5.6: MOS values and corresponding 95 % confidence intervals for reliable YouTube ratings in presence of α unreliable ratings. The unreliable user ratings follow a discrete uniform distribution as in Figure 5.5.

The three different curves for α in Figure 5.6 illustrate the following observations: Firstly, the obtained MOS values look all reasonable. The presentation of MOS values only does hence not allow to identify the validity and the reliability of the results. The results, however, clearly show a severe impact of unreliable users on the observed MOS values $\widetilde{R}_x$ for a test condition x. The true MOS value $\overline{R}_x$ for $\alpha = 0$ is shifted towards the average unreliable user rating $\overline{F} = 3$, with $\widetilde{R}_x = (1-\alpha)\overline{R}_x + \alpha\overline{F}$. Unfortunately, many subjective user studies only present MOS values without quantifying the reliability. Often confidence intervals are misused to quantify the reliability of the user ratings, but a 95 % confidence interval for a MOS value only shows that the mean rating including the unreliable user ratings lies within the confidence interval with a probability of 95 %. Secondly, the length of the confidence intervals I may therefore even decrease in the presence of unreliable user ratings due to the increased number N of ratings in total, as $I \sim 1/\sqrt{N}$. As a consequence, we conclude that reliability has to be quantified for QoE crowdtesting by appropriate means and that unreliable user ratings have to be filtered out which will be discussed in the next section.

Comparison of User Rating based Screening Mechanisms and Additional Reliability Mechanisms

In the literature there exist two overall categories of screening mechanisms: First, filtering of users based on the user ratings and second, screening of users independently of the ratings, but with additional reliability mechanisms, e.g., consistency tests [26, 158, 159]. For brevity, we will abbreviate *user rating based screening mechanism* with *URS* and *additional reliability mechanisms* with *ARM*. The ARM approach leads to extra effort in the implementation and in the analysis, however, unreliable users can be clearly identified. To illustrate this, we use the reliable and unreliable users from the earlier described YouTube experiments which follow the ARM approach and implement several reliability mechanisms from our proposed two-stage crowdtesting design as gold standard data. Then, we apply different URS screening mechanisms from literature and compare their ability to identify unreliable users for the YouTube results.

URS screening methods can be roughly separated into at least two classes [156]: One is based on finding inconsistencies compared with the mean result and relies on the ability of the subject to make correct identifications. The second class primarily removes subjects who cannot make the appropriate discriminations. Considering the variability of subjects' sensitivities to different artefacts [160], however, caution should be exercised in using URS [156]. As we will show in the following, URS screening mechanisms based on user ratings are not sufficient for QoE crowdtesting and thus ARM is necessary for unreliable user identification. Specifically, we will have a closer look the *ITU-R BT.500* [139], *crowdMOS* [88], *Random Clicker* [161], and *Quadrant of Euphoria* [142] as examples of URS screening mechanisms.

The *ITU-R BT.500* recommendation proposes to screen subjects with the β_2 test. It counts, whether the scores of subjects lie in an interval around the mean of all ratings for the corresponding test condition. The length of the interval above and below the mean is m-times the standard deviation of all ratings for this test condition. The kurtosis coefficient β_2 is then used to determine if the user scores are statistically normal or not. The factor m is chosen to be $m = 2$ (normal distribution) and $m = \sqrt{20}$ (no normal distribution), respectively. Based on this count, a user is rejected. More details can be found in [139]. Figure 5.7 shows the results for *ITU-R BT.500* (the very left group) and the remaining three URS screening mechanisms. Each screening mechanism is evaluated on the data of four different YouTube crowdtesting campaigns. It can be seen that only half of the users are filtered correctly by the β_2 test. It accepts, however, also a large ratio of unreliable users. Hence, this method alone is not recommended for QoE crowdtesting.

The *crowdMOS* framework for subjective user studies proposes a different URS screening mechanism: The sample correlation coefficient between the average user rating of a worker and the global average rating is used to identify unreliable users. The user ratings are averaged for the same test conditions, e.g., number of stalling events in the YouTube experiments. A user is rejected, if the correlation coefficient is below a certain threshold, e.g. 0.25 in [88]. Then the

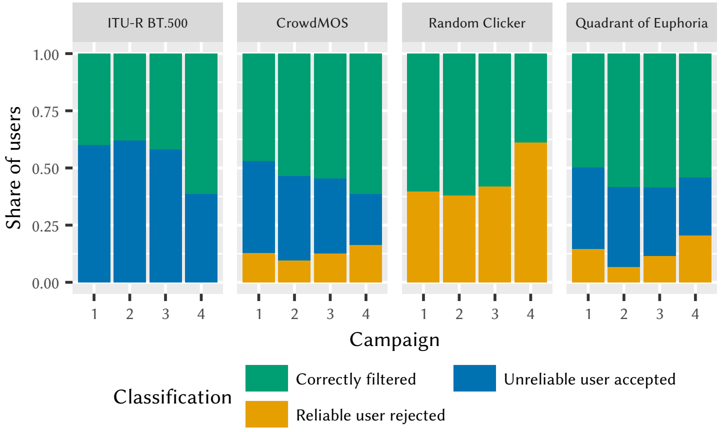


Figure 5.7: URS screening approaches filter out users based on their ratings. The application of those approaches to YouTube results shows that only a fraction of users is correctly identified. QoE crowdtesting requires ARM mechanisms.

global average rating is computed for the remaining users and the correlation coefficient is recomputed. Users are ranked in decreasing order of the correlation coefficient and the user screening starts again. Figure 5.7 shows that the crowdMOS approach filters only half of the users correctly. A large fraction of unreliable users is accepted, which can be reduced by increasing the threshold that, however, would result in an even larger ratio of reliable users rejected.

Kim et al. [161] investigate how to filter *Random Clickers* in a crowdsourcing-based study. In particular, Pearson's χ^2 test is applied to test the null hypothesis that the user is a random clicker. The resulting p-value is used for excluding users with a hight p-value above a certain threshold. Kim et al. use a threshold of 0.02. This approach seems quite promising, as the results shown previously clearly reveal unreliable user ratings to be randomly clicked. Figure 5.7, however, shows that the random clicker approach rejects many reliable user ratings, as they appear to be statistically random. This may be caused by the actual test design and the order of test conditions. Another explanation is the fact that users cannot differentiate the impact of the test conditions or perceive some test conditions equally. While an increased threshold reduces the ratio of rejected reliable users, it also leads to a higher ratio of accepted unreliable users. We conclude that the analysis of random user ratings is also not sufficient for unreliable user identification.

Chen et al. [142] present the *Quadrant of Euphoria* which is a web-based crowdsourcing platform for QoE assessment. For filtering out unreliable subjects, a new metric is introduced, the *Transitivity Satisfaction Rate* (TSR). TSR is defined as the number of judgement triplets satisfying transitivity divided by the total number of triplets where transitivity may apply. Transitivity in this context means the reasonable assumption that preferences of users are a transitive relation. Hence, if a user prefers the test condition A to B and B to C, the user will normally prefer A to C and users are rejected, if the TSR value is below 0.8 [142]. The obtained results are similar to the crowdMOS approach as a large ratio of unreliable users is accepted while some reliable users are rejected. Similarly to crowdMOS, the threshold value can be fine-tuned to reduce the acceptance of unreliable users, but at the cost of an increased rejection of reliable users.

We conclude that the URS approaches presented in this section and summarized in Table 5.4 cannot be used alone to clearly identify unreliable users. Hidden influence factors or the variability of subjects' sensitivities to different artefacts are not determined by those URS approaches. Although a combination of them may be interesting to improve screening, such as combining the random clicker approach and ITU-R BT.500, this remains a topic for future work. In summary, the screening of subjects should be done based on ARM methods as proposed in our two-stage design, which clearly identifies unreliable users independent of any hidden influence factor and the actual user rating. In [142], the payment of crowdsourcing users depends on a reliability metric in the form of the TSR above a certain threshold. Still, for the same reasons, it is recommended that payments are only refused when a subject is rejected according to the ARM methods.

5.2.5 Best Practices for QoE Crowdtesting

Summarising the rules proposed in the last sections in the form of best practices for designing QoE crowdtesting campaigns, we can differentiate between three main categories: *Technical implementation*, *campaign and task design*, and *statistical analysis.*

The *Technical implementation* of the test should take into consideration the spread of the used technology among the targeted crowd. The use of widely available technologies, e.g., HTML 5, are strongly recommended. Depending on required computational power, size of the crowd and/or geographical location, CDN networks and Cloud services can provide better service in comparison to a standalone server as discussed in Section 5.2.3. To cope with the limited reliability of the crowd and other factors influencing the rating behavior, for the *campaign and task design* we recommend the following steps:

Table 5.4: Comparison of URS screening methods based on user ratings.

Approach	Key measure for identifying reliable users
ITU-R BT.500 [139]	β_2 test counts the scores of a subject lying in an interval around the mean of all ratings for the corresponding test condition
CrowdMOS [88]	Sample correlation coefficient between the ratings of a subject and the global average rating
Random Clicker [161]	p-value of Pearson's χ^2 test that the user ratings are random
Quadrant of Euphoria [142]	Transitivity satisfaction rate by counting the triplets of user ratings u_A, u_B, u_C for conditions A, B, C following a transitive relation $(u_A < u_B \wedge u_B < u_C \Rightarrow u_A < u_C)$

1) The task should be designed to prevent cheating [75].

2) A pseudo-reliable crowd is created by simple, short, and cheap tests with different reliability elements. Only reliable users are then allowed to pass to the actual QoE tests with higher payments. This approach is also known as pilot task and main task [157].

3) Different elements need to be added in the task design to check the reliability of the users [26] and to filter out unreliable users in the first and second stage of the QoE test. Combining these elements also leads to an improved reliability of the results [162]. Additional ARM reliability mechanisms include, but are not limited to:

   a) Verification tests [158, 159], including captchas or computation of simple text equations: *"two plus 3=?"*, *"Which of these countries contains a major city called Cairo? (Brazil, Canada, Egypt, Japan)"*.
   b) Consistency tests: First, the user is asked *"In which country do you live?"*. Later, the user is asked *"On which continent do you live?"*.
   c) Content questions about the test: *"Which animal did you see?" (Lion, Bird, Rabbit, Fish)*.
   d) Gold standard data [163]: *"Did you notice any stops to the video you just watched?" (Yes, No)*, when the actual test video did not include any stallings.
   e) Application-layer monitoring [20, 26]: Monitoring of response times of users and browser events to capture the focus time.

The important thing to keep in mind is not to add too many reliability items, as otherwise the assessment task will become too lengthy. Further, too many of these questions may give a signal of distrust to the users. As a result, users may abort the survey. In general, incentives and proper payment schemes depending on the actual work effort are the key to high quality work. Incentive schemes such as gamification have the potential to make crowdsourcing an even more powerful tool to deliver high data quality in QoE assessment.

Regarding the best practices for evaluating the campaign and calculating the overall *statistics* of the crowdsourcing testing we encourage the use of a combination of URS and ARM mechanisms. URS methods alone cannot clearly identify unreliable users, since, e.g., hidden influence factors or the variability of subjects' sensitivities to different artefacts are not determined. We therefore recommend to use ARM approaches for screening of test subjects that are able to clearly identify unreliable users independent of any hidden influence factor and the actual user rating. Nevertheless, reliability measures such as inter- and intrarater reliability should always be given for QoE crowdtesting studies, where high values show reliable user ratings, but low values imply the presence of unreliable users or hidden influence factors in the QoE crowdtesting campaign [3]. The results of crowdsourcing user studies should always be accompanied by the description of the used ARM mechanisms, and also by the description of other reliability measures which have been used.

5.3 Lessons Learned

In this chapter we presented two use cases illustrating the benefit of using crowdsourcing in scientific research. We showed that crowdsourcing based network measurements can be used to achieve a realistic view on the network from an end-user perspective. In comparison to existing measurement approaches, crowdsourcing based measurements provide a) a less detailed, but also less biased view than data provided by Internet service providers, b) a less reliable, but more diverse and realistic view than measurements in distributed testbeds, c) slight more costly, but much faster measurements than voluntary participation approaches. However, traditional measurement setups cannot be crowdsourced without adaptations incorporating the technical limitations of the end-user devices and other crowdsourcing specific requirements, like additional quality assurance mechanisms.

Similarly, we demonstrated that Quality of Experience tests can be crowdsourced enabling large scale assessments in realistic environments. As Quality of experience tests are based on subjective ratings for the participants, most common crowdsourcing quality control mechanisms like gold standard data are inapplicable. Also common quality control techniques used in laboratory setting cannot be applied directly. However, we showed that using a two-stage campaign design in conjunction with a combination of different quality control mechanisms, e.g., content questions, consistency questions, and application layer monitoring, allows to obtain reliable results in a timely and cost effective manner.

6 Conclusions

Computer systems have evolved from large scale machines used for calculations, to handheld devices that can almost act as personal assistants. But even if computer systems replace human work-force in many parts of everyday life, there still exists a large number of tasks that cannot be automated, yet. This also includes tasks, which we consider to be rather simple like the categorization of image content or subjective ratings. Traditionally, these tasks have been completed by designated employees or outsourced to specialized companies. Recently the crowdsourcing paradigm is more and more applied to complete human-labor intensive tasks. Crowdsourcing aims at leveraging the huge number of Internet users all around the globe, which form a potentially highly available, low-cost, and easy accessible work-force.

To enable the distribution of work on a global scale, new web-based services emerged, so called crowdsourcing platforms, that act as mediator between employers posting tasks and workers completing tasks. However, the crowdsourcing approach, especially the large anonymous worker crowd, results in two types of challenges. On the one hand, there are technical challenges like the dimensioning of crowdsourcing platform infrastructure or the interconnection of crowdsourcing platforms and machine clouds to build hybrid services. On the other hand, there are conceptual challenges like identifying reliable workers or migrating traditional off-line work to the crowdsourcing environment.

In this monograph, we analyze and model current crowdsourcing systems to optimize crowdsourcing workflows and the underlying infrastructure. We review existing crowdsourcing systems and use the resulting categorization of crowdsourcing tasks and platforms to derive generalizable properties. Based on this categorization and an analysis of a commercial crowdsourcing platform, we develop models for different aspects of crowdsourcing platforms and crowdsourcing mechanisms. A special focus is put on quality assurance mechanisms for crowdsourcing tasks, where the developed models are used to assess the suitability and costs of existing approaches for different types of tasks. Further, a novel quality assurance mechanism is developed and its feasibility is shown. The findings from the analysis of existing platforms, the derived models, and the developed quality assurance mechanisms are finally used to derive best practises for two crowdsourcing use-cases. These two exemplary use-cases cover aspects typical for a large range of crowdsourcing tasks and illustrated the potential benefits, but also resulting challenges when using crowdsourcing.

The analysis of current crowdsourcing systems presented in this monograph consist of two major parts. In a first step, we review the general concepts of the crowdsourcing approach and how they are implemented by today's crowdsourcing platform providers. This leads to a coarse categorization of tasks and platforms, which can be used for system modeling and helps to identify common properties of a large number of tasks and platforms. In a second step, we provide an in-depth analysis of a single commercial platform that is based on a database snapshot from Microworkers.com. The evaluation shows that the users have a heterogeneous cultural background. This has a significant impact on how crowdsourcing tasks need to be designed by employers to achieve optimal results. Due to the different native languages of the crowdsourcing workers, multi-lingual task interfaces might be required to leverage the full potential of the workers. Further, cultural differences might affect task ratings, e.g., while identifying mature content.

The results of the analysis of crowdsourcing systems are then used to model the growth of crowdsourcing platforms and the activity of the users. We show that even a simple square growth model enables platform operators to accurately estimate the future development of the user base for up to two years. These estimations can be used to dimension the platform infrastructure accordingly. Moreover, a fluid model is presented describing the registration process of platform users and the transition between active and inactive users on the platform. This model enables the platform operator to evaluate different advertising strategies to attract new users and also different strategies to keep registered users active. Comparing different advertisement campaign lengths and durations, we identify short and high intense campaigns as the most promising approach in this context.

One of the most challenging aspects of crowdsourcing and also one of the most important factors from an employer's point of view is the quality control of task results submitted by the workers. A common approach is assessing a worker's reliability and use this information to estimate the quality of the submitted results, like in the gold standard approach. However, this often results in higher costs per task, due to the additional efforts for the workers while performing the reliability tests. To overcome this, we illustrated the possibility to assess a worker's reliability based on his interactions with the task interface. Using a poof-of-concept implementation, we show that this approach is capable of identifying unreliable workers without additional consistency or gold standard questions. Further, we develop models for a majority decision and a control group based quality assurance mechanisms that allow to assess the capability of detecting invalid results and the corresponding costs. From our evaluation we can conclude that majority decisions are cost effective for simple and cheap tasks, while a quality assurance mechanism using a control group is cost-effective for high paid tasks.

In the final part of this monograph, we illustrate the benefits of crowdsourcing for network measurements and subjective user studies, but also potential pitfalls and limitations. Using a comprehensive set of different studies, we are able to derive a set of best practises for designing tasks and adapting existing research methodology to the crowdsourcing environment. Due to the diverse requirements of the crowdsourced network measurements and the crowdsourced subjective studies, the derived guidelines can easily be applied to a wide range of commercial crowdsourcing applications, as well.

With the ongoing digitalization and globalization of the labor markets, the crowdsourcing paradigm is expected to gain even more importance in the next years. This is already evident in the currently new emerging fields of crowdsourcing, like enterprise crowdsourcing or mobile crowdsourcing. The models developed in this monograph enable platform providers to optimize their current systems and employers to optimize their workflows to increase their commercial success. Moreover, the work presented improves the general understanding of crowdsourcing systems, a key for identifying necessary adaptions and future improvements.

Bibliography and References

Bibliography of the Author

Journal Papers

[1] M. Hirth, T. Hoßfeld, M. Mellia, C. Schwartz, and F. Lehrieder, “Crowdsourced Network Measurements: Benefits and Best Practices”, *Computer Networks*, vol. 90, Special Issue on Crowdsourcing Oct. 2015.

[2] T. Hoßfeld, V. Burger, H. Hinrichsen, M. Hirth, and P. Tran-Gia, “On the Computation of Entropy Production in Stationary Social Networks”, *Social Network Analysis and Mining*, vol. 4, Apr. 2014.

[3] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, and P. Tran-Gia, “Best Practices for QoE Crowdtesting: QoE Assessment with Crowdsourcing”, *Transactions on Multimedia*, vol. 16, no. 2, Feb. 2014.

[4] T. Hoßfeld, M. Hirth, and P. Tran-Gia, “Crowdsourcing - Modell einer neuen Arbeitswelt im Internet”, *Wirtschaftsinformatik & Management*, vol. 5, Oct. 2013.

[5] P. Tran-Gia, T. Hoßfeld, M. Hartmann, and M. Hirth, “Crowdsourcing and its Impact on Future Internet Usage”, *IT - Information Technology*, vol. 55, no. 4, Jul. 2013.

[6] T. Hoßfeld, M. Hirth, and P. Tran-Gia, “Aktuelles Schlagwort: Crowdsourcing”, *Informatik Spektrum*, vol. 35, no. 3, Apr. 2012.

[7] M. Hirth, T. Hossfeld, and P. Tran-Gia, “Analyzing Costs and Accuracy of Validation Mechanisms for Crowdsourcing Platforms”, *Mathematical and Computer Modelling*, vol. 57, no. 11-12, Jun. 2012.

[8] B. Staehle, F. Wamser, M. Hirth, D. Stezenbach, and D. Staehle, “AquareYoum: Application and Quality of Experience-Aware Resource Management for YouTube in Wireless Mesh Networks”, *PIK - Praxis der Informationsverarbeitung und Kommunikation*, vol. 34, no. 3, Aug. 2011.

Conference Papers

[9] C. Schwartz, K. Borchert, M. Hirth, and P. Tran-Gia, “Modeling Crowdsourcing Platforms to Enable Workforce Dimensioning”, in *Proceedings of the International Telecommunication Networks and Applications Conference*, Sydney, Australia, Nov. 2015.

[10] M. Becker, K. Borchert, M. Hirth, H. Mewes, A. Hotho, and P. Tran-Gia, “MicroTrails: Comparing Hypotheses about Task Selection on a Crowd Sourcing Platform”, in *Proceedings of the International Conference on Knowledge Technologies and Data-driven Business*, Graz, Austria, Oct. 2015.

[11] P. Lebreton, I. H. Torres, T. Mäki, E. Skodras, and M. Hirth, “Eye Tracker in the Wild: Studying the Delta Between What is Said and Measured in a Crowdsourcing Experiment”, in *Proceedings of the Workshop on Crowdsourcing for Multimedia*, Brisbane, Australia, Oct. 2015.

[12] T. Zinner, F. Lemmerich, S. Schwarzmann, M. Hirth, P. Karg, and A. Hotho, “Text Categorization for Deriving the Application Quality in Enterprises using Ticketing Systems”, in *Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery*, Valencia, Spain, Sep. 2015.

[13] L. Dinh-Xuan, C. Schwartz, M. Hirth, F. Wamser, and H. T. Thu, "Analyzing the Impact of Delay and Packet Loss on Google Docs", in *Proceedings of the International Conference on Mobile Networks and Management*, Santander, Spain, Sep. 2015.

[14] S. Schnitzer, C. Rensing, S. Schmidt, K. Borchert, M. Hirth, and P. Tran-Gia, "Demands on Task Recommendation in Crowdsourcing Platforms - The Worker's Perspective", in *Proceedings of the Workshop on Crowdsourcing and Human Computation for Recommender Systems*, Vienna, Austria, Sep. 2015.

[15] P. Lebreton, E. Skodras, T. Mäki, I. H. Torres, and M. Hirth, "Bridging the Gap Between Eye Tracking and Crowdsourcing", in *Proceedings of the SPIE 9394, Human Vision and Electronic Imaging XX*, San Francisco, California, USA, Feb. 2015.

[16] C. Schwartz, M. Hirth, T. Hoßfeld, and P. Tran-Gia, "Performance Model for Waiting Times in Cloud File Synchronization Services", in *Proceedings of the International Teletraffic Congress*, Karlskrona, Sweden, Sep. 2014.

[17] T. Hoßfeld, M. Hirth, P. Korshunov, P. Hanhart, B. Gardlo, C. Keimel, and C. Timmerer, "Survey of Web-based Crowdsourcing Frameworks for Subjective Quality Assessment", in *Proceedings of the International Workshop on Multimedia Signal Processing*, Jakarta, Indonesia, Sep. 2014.

[18] S. Schwarzmann, T. Zinner, and M. Hirth, "Deriving the Employee-perceived Application Quality in Enterprise IT Infrastructures using Information from Ticketing Systems", in *Proceedings of the Workshop on Knowledge Discovery, Data Mining and Machine Learning*, Aachen, Germany, Sep. 2014.

[19] I. Hupont, P. Lebreton, T. Mäki, E. Skodras, and M. Hirth, "Is affective crowdsourcing reliable?", in *Proceedings of the International Conference on Communications and Electronics*, Da Nang, Vietnam, Jul. 2014.

[20] M. Hirth, S. Scheuring, T. Hoßfeld, C. Schwartz, and P. Tran-Gia, “Predicting Result Quality in Crowdsourcing Using Application Layer Monitoring”, in *Proceedings of the International Conference on Communications and Electronics*, Da Nang, Vietnam, Jul. 2014.

[21] V. Burger, M. Hirth, C. Schwartz, and T. Hoßfeld, “Increasing the Coverage of Vantage Points in Distributed Active Network Measurements by Crowdsourcing”, in *Proceedings of Measurement, Modelling and Evaluation of Computing Systems*, Bamberg, Germany, Mar. 2014.

[22] M. Seufert, K. Lorey, M. Hirth, and T. Hoßfeld, “Gamification Framework for Personalized Surveys on Relationships in Online Social Networks”, in *Proceedings of the International Workshop on Crowdsourcing and Gamification in the Cloud*, Dresden, Germany, Dec. 2013.

[23] P. Amrehn, K. Vandenbroucke, T. Hoßfeld, K. de Moor, M. Hirth, R. Schatz, and P. Casas, “Need for Speed? On Quality of Experience for File Storage Services”, in *Proceedings of the International Workshop on Perceptual Quality of Systems*, Vienna, Austria, Sep. 2013.

[24] H. Hinrichsen, T. Hoßfeld, M. Hirth, and P. Tran-Gia, “Entropy Production in Stationary Social Networks”, in *Proceedings of the Workshop on Complex Networks*, Berlin, Germany, Mar. 2013.

[25] M. Hirth, F. Lehrieder, S. Oberste-Vorth, T. Hoßfeld, and P. Tran-Gia, “Wikipedia and its Network of Authors from a Social Network Perspective”, in *Proceedings of the International Conference on Communications and Electronics*, Hue, Vietnam, Aug. 2012.

[26] T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia, “Quantification of YouTube QoE via Crowdsourcing”, in *Proceedings of the International Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions*, Dana Point, California, USA, Dec. 2011.

[27] T. Hoßfeld, M. Hirth, and P. Tran-Gia, “Modeling of Crowdsourcing Platforms and Granularity of Work Organization in Future Internet”, in *Proceedings of the International Teletraffic Congress*, San Francisco, California, USA, Sep. 2011.

[28] M. Hirth, T. Hoßfeld, and P. Tran-Gia, “Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com”, in *Proceedings of the Workshop on Future Internet and Next Generation Networks*, Seoul, Korea, Jun. 2011.

[29] ——, “Cost-Optimal Validation Mechanisms and Cheat-Detection for Crowdsourcing Platforms”, in *Proceedings of the Workshop on Future Internet and Next Generation Networks*, Seoul, Korea, Jun. 2011.

[30] B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle, “Aquarema in Action: Improving the YouTube QoE in Wireless Mesh Networks”, in *Proceedings of the Baltic Congress on Future Internet Communications*, Riga, Latvia, Feb. 2011.

[31] ——, “YoMo: A YouTube Application Comfort Monitoring Tool”, in *Proceedings of New Dimensions in the Assessment and Support of Quality of Experience for Multimedia Applications*, Tampere, Finland, Jun. 2010.

[32] M. Hirth, B. Staehle, F. Wamser, R. Pries, and D. Staehle, “QoE Prediction for Radio Resource Management”, in *Proceedings of the International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, Berlin, Germany, May 2010.

[33] B. Staehle, D. Staehle, R. Pries, M. Hirth, A. Kassler, and P. Dely, “Measuring One-Way Delay in Wireless Mesh Networks - An Experimental Investigation”, in *Proceedings of the Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, Tenerife, Canary Islands, Spain, Oct. 2009.

White Papers

[34] T. Hoßfeld, M. Hirth, J. Redi, F. Mazza, P. Korshunov, B. Naderi, M. Seufert, B. Gardlo, S. Egger, and C. Keimel, *Best Practices and Recommendations for Crowdsourced QoE - Lessons Learned from the Qualinet Task Force Crowdsourcing*, Oct. 2014.

General References

[35] J. Howe, "The Rise of Crowdsourcing", *Wired Magazine*, vol. 14, no. 6, Jun. 2006.

[36] U. Gadiraju, R. Kawase, and S. Dietze, "A Taxonomy of Microtasks on The Web", in *Proceedings of the Conference on Hypertext and Social Media*, Santiago, Chile, Sep. 2014.

[37] M. A. AlShehry and B. W. Ferguson, "A Taxonomy of Crowdsourcing Campaigns", in *Proceedings of the International Conference on World Wide Web*, Florence, Italy, May 2015.

[38] E. Schenk and C. Guittard, "Towards a Characterization of Crowdsourcing Practices", *Journal of Innovation Economics & Management*, vol. 1, no. 7, 2011.

[39] Amazon Mechanical Turk. (Mar. 2016), [Online]. Available: http://mturk.com.

[40] Microworkers. (Mar. 2016), [Online]. Available: http://microworkers.com.

[41] Streetspotr. (Mar. 2016), [Online]. Available: http://www.streetspotr.com.

[42] InnoCentive. (Mar. 2016), [Online]. Available: http://www.innocentive.com.

[43] 99design. (Mar. 2016), [Online]. Available: http://www.99designs.com.

[44] CrowdFlower. (Mar. 2016), [Online]. Available: http://www.crowdflower.com.

[45] CrowdSource. (Mar. 2016), [Online]. Available: http://www.crowdsource.com.

[46] Microtask. (Mar. 2016), [Online]. Available: http://www.microtask.com.

[47] TaskRabbit. (Mar. 2016), [Online]. Available: http://www.taskrabbit.com.

[48] Facebook. (Mar. 2016), [Online]. Available: http://www.facebook.com.

[49] Cisco VNI, "Forecast and Methodology, 2014-2019", Technical Report, Cisco, Tech. Rep., May 2015.

[50] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-Based Character Recognition via Web Security Measures", *Science*, vol. 321, no. 5895, Sep. 2008.

[51] reCaptcha. (Mar. 2016), [Online]. Available: http://www.google.com/recaptcha.

[52] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently Scaling up Crowdsourced Video Annotation", *International Journal of Computer Vision*, vol. 101, no. 1, Jan. 2013.

[53] C. Keimel, J. Habigt, C. Horch, and K. Diepold, "Qualitycrowd - A Framework for Crowd-Based Quality Evaluation", in *Proceedings of the Picture Coding Symposium*, Krakow, Poland, May 2012.

[54] J. Ross, L. Irani, M. Silberman, A. Zaldivar, and B. Tomlinson, "Who are the Crowdworkers?: Shifting Demographics in Mechanical Turk", in *Proceedings of the Conference on Human Factors in Computing Systems*, Atlanta, Georgia, USA, Apr. 2010.

[55] F. Lehrieder, G. Dán, T. Hoßfeld, S. Oechsner, and V. Singeorzan, "Caching for BitTorrent-like P2P Systems: A Simple Fluid Model and its Implications", *Transactions on Networking*, vol. 20, no. 4, Aug. 2012.

[56] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the Intercloud-Protocols and Formats for Cloud Computing Interoperability", in *Proceedings of the International Conference on Internet and Web Applications and Services*, Venice/Mestre, Italy, May 2009.

[57] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-Datacenter Bulk Transfers with Netstitcher", vol. 41, no. 4, Aug. 2011.

[58] T. Vander Wal. (Feb. 2007). Folksonomy, [Online]. Available: http://vanderwal.net/folksonomy.html.

[59] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From Game Design Elements to Gamefulness: Defining Gamification", in *Proceedings of the International Academic MindTrek Conference: Envisioning Future Media Environments*, Tampere, Finnland, Sep. 2011.

[60] M. S. Bernstein, D. R. Karger, R. C. Miller, and J. Brandt, "Analytic Methods for Optimizing Realtime Crowdsourcing", in *Proceedings of the Collective Intelligence Conference*, Cambrige, Massachusetts, USA, Apr. 2012.

[61] J. Redi, T. Hoßfeld, P. Korshunov, F. Mazza, I. Povoa, and C. Keimel, "Crowdsourcing-Based Multimedia Subjective Evaluations: A Case Study on Image Recognizability and Aesthetic Appeal", in *Proceedings of the CrowdMM Workshop*, Barcelona, Spain, Oct. 2013.

[62] T. Volk, C. Keimel, M. Moosmeier, and K. Diepold, "Crowdsourcing vs. Laboratory Experiments - QoE Evaluation of Binaural Playback in a Teleconference Scenario", *Computer Networks*, vol. 90, Special Issue on Crowdsourcing.

[63] D. Ghadiyaram and A. C. Bovik, "Massive Online Crowdsourced Study of Subjective and Objective Picture Quality", *Transactions on Image Processing*, vol. 25, no. 1, Jan. 2016.

[64] P. Ipeirotis. (Mar. 2008). Mechanical Turk: The Demographics, [Online]. Available: http://behind-the-enemy-lines.blogspot.com/2008/03/mechanical-turk-demographics.html.

[65] ——, (Mar. 2009). Turker Demographics vs. Internet Demographics, [Online]. Available: http://behind-the-enemy-lines.blogspot.com/2009/03/turker-demographics-vs-internet.html.

[66] ——, (Mar. 2010). The New Demographics of Mechanical Turk, [Online]. Available: http://behind-the-enemy-lines.blogspot.com/2010/03/new-demographics-of-mechanical-turk.html.

[67] J. Ross, A. Zaldivar, L. Irani, and B. Tomlinson, "Who are the Turkers? Worker Demographics in Amazon Mechanical Turk", Department of Informatics, University of California, Irvine, USA, Tech. Rep. SocialCode-2009-01, Jan. 2009.

[68] P. Ipeirotis, "Analyzing the Amazon Mechanical Turk Marketplace", *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 2, 2010.

[69] United Nations Development Programme, *Human Development Report 2015*. Selim Jahan, 2015. [Online]. Available: http://hdr.undp.org/sites/default/files/2015_human_development_report_1.pdf.

[70] S. Faradani, B. Hartmann, and P. G. Ipeirotis, "What's the Right Price? Pricing Tasks for Finishing on Time", in *Proceedings of the Workshop on Human Computation*, San Francisco, California, USA, Aug. 2011.

[71] J. Wang, S. Faridani, and P. G. Ipeirotis, "Estimating the Completion Time of Crowdsourced Tasks Using Survival Analysis Models", in *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining*, Hong Kong, China, Feb. 2011.

[72] J. J. Moré, "The Levenberg-Marquardt Algorithm: Implementation and Theory", in *Numerical Analysis*, Springer, 1978, pp. 105–116.

[73] S. Suri, D. Goldstein, and W. Mason, “Honesty in an Online Labor Market”, in *Proceedings of the Workshop on Human Computation*, San Francisco, California, USA, Aug. 2011.

[74] S. Khanna, A. Ratan, J. Davis, and W. Thies, “Evaluating and Improving the Usability of Mechanical Turk for Low-income workers in India”, in *Proceedings of the Symposium on Computing for Development*, London, United Kingdom, Dec. 2010.

[75] A. Kittur, E. H. Chi, and B. Suh, “Crowdsourcing User Studies with Mechanical Turk”, in *Proceeding of the Conference on Human Factors in Computing Systems*, Florence, Italy, Apr. 2008.

[76] C. Eickhoff and A. P. de Vries, “Increasing Cheat Robustness of Crowdsourcing Tasks”, *Information Retrieval*, vol. 16, no. 2, Apr. 2012.

[77] C. Eickhoff and A. de Vries, “How Crowdsourcable is Your Task?”, in *Proceedings of the Workshop on Crowdsourcing for Searchand Data Mining*, Hong Kong, China, Feb. 2011.

[78] U. Gadiraju, P. Siehndel, B. Fetahu, and R. Kawase, “Breaking Bad: Understanding Behavior of Crowd Workers in Categorization Microtasks”, in *Proceedings of the Conference on Hypertext and Social Media*, Nicosia, Northern Cyprus, Sep. 2015.

[79] J. Le, A. Edmonds, V. Hester, and L. Biewald, “Ensuring Quality in Crowdsourced Search Relevance Evaluation: The Effects of Training Question Distribution”, in *Proceedings of the Workshop on Crowdsourcing for Search Evaluation*, Geneva, Switzerland, Jul. 2010.

[80] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, and L. Biewald, “Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing”, in *Proceedings of the Workshop on Human Computation Workshop*, San Francisco, California, USA, Aug. 2011.

[81] P. Ipeirotis, F. Provost, and J. Wang, "Quality Management on Amazon Mechanical Turk", in *Proceedings of the Workshop on Human Computation*, Washington DC, USA, Jul. 2010.

[82] K. Chen, C. Chang, C. Wu, Y. Chang, C. Lei, and C. Sinica, "Quadrant of Euphoria: A Crowdsourcing Platform for QoE Assessment", *Network*, vol. 24, no. 2, Mar. 2010.

[83] G. Kazai, J. Kamps, and N. Milic-Frayling, "Worker Types and Personality Traits in Crowdsourcing Relevance Labels", in *Proceedings of the International Conference on Information and Knowledge Management*, Glasgow, United Kingdom, Oct. 2011.

[84] U. Gadiraju, R. Kawase, S. Dietze, and G. Demartini, "Understanding Malicious Behavior in Crowdsourcing Platforms: The Case of Online Surveys", in *Proceedings of Conference on Human Factors in Computing Systems*, Seoul, Republic of Korea, Apr. 2015.

[85] J. M. Rzeszotarski and A. Kittur, "Instrumenting the Crowd: Using Implicit Behavioral Measures to Predict Task Performance", in *Proceedings of the Symposium on User Interface Software and Technology*, Santa Barbara, California, USA, Oct. 2011.

[86] R. R. Day and J.-S. Park, "Developing reading comprehension questions", *Reading in a Foreign Language*, vol. 17, no. 1, Apr. 2005.

[87] A. Kapelner and D. Chandler, "Preventing Satisficing in Online Surveys", in *Proceedings of the Conference about Crowdsourcing*, San Francisco, California, USA, Oct. 2010.

[88] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "CrowdMOS: An Approach for Crowdsourcing Mean Opinion Score Studies", in *Proceedings of the Conference on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, May 2011.

[89] W. Mason and S. Suri, "Conducting Behavioral Research on Amazon's Mechanical Turk", *Behavior Research Methods*, vol. 44, no. 1, Mar. 2012.

[90] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux, "Mechanical Cheat: Spamming Schemes and Adversarial Techniques on Crowdsourcing Platforms", in *Proceedings of the Workshop on Crowdsourcing Web Search*, Lyon, FR, Apr. 2012.

[91] L. Von Ahn and L. Dabbish, "Labeling Images with a Computer Game", in *Proceedings of the Conference on Human Factors in Computing Systems*, Vienna, Austria, Apr. 2004.

[92] G. Little, L. Chilton, M. Goldman, and R. Miller, "Turkit: Tools for Iterative Tasks on Mechanical Turk", in *Proceedings of the Workshop on Human Computation*, Paris, France, Jun. 2009.

[93] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Exploring Iterative and Parallel Human Computation Processes", in *Proceedings of the Workshop on Human Computation*, Washington DC, USA, Jul. 2010.

[94] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich, "Soylent: A Word Processor with a Crowd Inside", in *Proceedings of the Symposium on User Interface Software and Technology*, New York, New York, USA, Oct. 2010.

[95] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "Crowdforge: Crowdsourcing Complex Work", in *Proceedings of the Symposium on User Interface Software and Technology*, Santa Barbara, California, USA, Oct. 2011.

[96] A. Kulkarni, M. Can, and B. Hartmann, "Collaboratively Crowdsourcing Workflows with Turkomatic", in *Proceedings of the Conference on Computer Supported Cooperative Work*, Seattle, Washington, USA, Feb. 2012.

[97] S. Dow, A. Kulkarni, B. Bunge, T. Nguyen, S. Klemmer, and B. Hartmann, "Shepherding the Crowd: Managing and Providing Feedback to Crowd Workers", in *Proceedings of the Conference on Human Factors in Computing Systems*, Vancover, British Columba, Canada, May 2011.

[98] A. Kittur and R. Kraut, “Harnessing the Wisdom of Crowds in Wikipedia: Quality Through Coordination”, in *Proceedings of the Conference on Computer Supported Cooperative Work*, San Diego, California, USA, Nov. 2008.

[99] D. Iren and S. Bilgen, “Cost of Quality in Crowdsourcing”, *Human Computation*, vol. 1, no. 2, 2014.

[100] G. Kazai and I. Zitouni, “Quality Management in Crowdsourcing Using Gold Judges Behavior”, in *Proceedings of the International Conference on Web Search and Data Mining*, San Francisco, California, USA, Feb. 2016.

[101] S. Zogaj, U. Bretschneider, and J. M. Leimeister, “Managing crowdsourced software testing: A case study based insight on the challenges of a crowdsourcing intermediary”, *Journal of Business Economics*, vol. 84, no. 3, Feb. 2014.

[102] Q. Su, D. Pavlov, J.-H. Chow, and W. C. Baker, “Internet-Scale Collection of Human-Reviewed Data”, in *Proceedings of the International Conference on World Wide Web*, Banff, Alberta, Canada, May 2007.

[103] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng, “Cheap and Fast - But is it Good?: Evaluating Non-Expert Annotations for Natural Language Tasks”, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawai, USA, Oct. 2008.

[104] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, *et al.*, “Galaxy Zoo: Morphologies Derived from Visual Inspection of Galaxies from the Sloan Digital Sky Survey”, *Monthly Notices of the Royal Astronomical Society*, vol. 389, no. 3, Sep. 2008.

[105] L. Maier-Hein, S. Mersmann, D. Kondermann, C. Stock, H. G. Kenngott, A. Sanchez, M. Wagner, A. Preukschas, A.-L. Wekerle, S. Helfert, *et al.*, “Crowdsourcing for Reference Correspondence Generation in En-

doscopic Images", in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, Boston, Massachusetts, USA, Sep. 2014.

[106] S. M. Smith, C. A. Roster, L. L. Golden, and G. S. Albaum, "A Multi-Group Analysis of Online Survey Respondent Data Quality: Comparing a Regular USA Consumer Panel to MTurk Samples", *Journal of Business Research*, 2016.

[107] D. L. Crone and L. A. Williams, "Crowdsourcing Participants for Psychological Research in Australia: A Test of Microworkers", *Australian Journal of Psychology*, 2016.

[108] V. K. Adhikari, S. Jain, and Z.-L. Zhang, "YouTube Traffic Dynamics and its Interplay with a Tier-1 ISP: An ISP Perspective", in *Proceedings of the SIGCOMM Conference*, New Delhi, India, Aug. 2010.

[109] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, "Anatomy of a Large European IXP", in *Proceedings of the SIGCOMM Conference*, Helsinki, Finland, Aug. 2012.

[110] A. Finamore, M. Mellia, M. Meo, M. Munafo, and D. Rossi, "Experiences of Internet Traffic Monitoring with tstat", *IEEE Network*, vol. 25, no. 3, May 2011.

[111] P. M. Santiago del Rio, D. Rossi, F. Gringoli, L. Nava, L. Salgarelli, and J. Aracil, "Wire-Speed Statistical Classification of Network Traffic on Commodity Hardware", in *Proceedings the of Internet Measurement Conference*, Boston, Massachusetts, USA, Nov. 2012.

[112] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services", *Computer Communication Review*, vol. 33, no. 3, Jun. 2003.

[113] C. Dovrolis, K. Gummadi, A. Kuzmanovic, and S. D. Meinrath, "Measurement Lab: Overview and an Invitation to the Research Community", *Computer Communication Review*, vol. 40, no. 3, Jul. 2010.

[114] GENI. (Mar. 2016), [Online]. Available: http://www.geni.net/.

[115] GLab. (Mar. 2016), [Online]. Available: http://www.german-lab.de.

[116] N. Spring, L. Peterson, A. Bavier, and V. Pai, “Using PlanetLab for Network Research: Myths, Realities, and Best Practices”, *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, Jun. 2006.

[117] Y. Shavitt and E. Shir, “DIMES: Let the Internet Measure Itself”, *Computer Communications Review*, vol. 35, no. 5, Oct. 2005.

[118] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, “iPlane: An Information Plane for Distributed Services”, in *Proceedings of the Symposium on Operating Systems Design and Implementation*, Seattle, Washington, USA, Nov. 2006.

[119] Z. Wen, S. Triukose, and M. Rabinovich, “Facilitating Focused Internet Measurements”, in *Proceedings of the SIGMETRICS Conference*, San Diego, California, USA, Jun. 2007.

[120] M. Dhawan, J. Samuel, R. Teixeira, C. Kreibich, M. Aollman, N. Weaver, and V. Paxson, “Fathom: A Browser-Based Network Measurement Platform”, in *Proceedings of Internet Measurement Conference*, Boston, Massachusettes, USA, Nov. 2012.

[121] D. R. Choffnes, F. E. Bustamante, and Z. Ge, “Crowdsourcing Service-Level Network Event Monitoring”, in *Proceedings of the SIGCOMM Conference*, New Delhi, India, Aug. 2010.

[122] M. A. Sánchez, J. S. Otto, Z. S. Bischof, D. R. Choffnes, F. E. Bustamante, B. Krishnamurthy, and W. Willinger, “DASU: Pushing Experiments to the Internet’s Edge”, in *Proceedings of the Symposium on Networked Systems Design and Implementation*, Lombard, Illinois, USA, Apr. 2013.

[123] SamKnows. (Mar. 2016), [Online]. Available: http://www.samknows.com/.

[124] Ripe NCC. (Mar. 2016), [Online]. Available: http://atlas.ripe.net/.

[125] Conviva. (Mar. 2016), [Online]. Available: http://www.conviva.com.

[126] D. R. Choffnes and F. E. Bustamante, “Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems”, *Computer Communications Review*, vol. 38, no. 4, Aug. 2008.

[127] K. Thompson, G. J. Miller, and R. Wilder, “Wide-Area Internet Traffic Patterns and Characteristics”, *Network*, vol. 11, no. 6, Aug. 1997.

[128] R. Van De Meent, “Network Link Dimensioning: A Measurement & Modeling Based Approach”, PhD thesis, Centre of Telematics and Information Technology, University of Twente, 2006.

[129] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, “Drafting Behind Akamai (Travelocity-Based Detouring)”, in *Proceedings of the SIGCOMM Conference*, Pisa, Italy, Sep. 2006.

[130] S. Banerjee, T. G. Griffin, and M. Pias, “The Interdomain Connectivity of PlanetLab Nodes”, in *Proceedings of the Workshop on Passive and Active Network Measurement*, Juan-les-Pins, France, Apr. 2004.

[131] Speedtest. (Mar. 2016), [Online]. Available: http://www.speedtest.net.

[132] Matt Martz. (Mar. 2016), [Online]. Available: http://github.com/sivel/speedtest-cli.

[133] M. Varela, T. Mäki, L. Skorin-Kapov, and T. Hoßfeld, “Increasing Payments in Crowdsourcing: Don’t Look a Gift Horse in the Mouth”, in *Proceedings of the Workshop on Perceptual Quality of Systems*, Vienna, Austria, Sep. 2013.

[134] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras, “Inside Dropbox: Understanding Personal Cloud Storage Services”, in *Proceedings of the Internet Measurement Conference*, Boston, Massachusetts, USA, Nov. 2012.

[135] D. Vakharia and M. Lease, “Beyond Mechanical Turk: An Analysis of Paid Crowd Work Platforms”, in *Proceedings of the iConference*, Newport Beach, California, USA, Mar. 2015.

[136] R. K. Mok, W. Li, and R. K. Chang, “A User Behavior Based Cheat Detection Mechanism for Crowdtesting”, in *Proceedings of the SIGCOMM Conference*, Chicago, Illinois, USA, Aug. 2014.

[137] P. L. Callet, S. Möller, and A. Perkis (eds), *Qualinet White Paper on Definitions of Quality of Experience*, Lausanne, Switzerland, Jun. 2012.

[138] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, “Internet video delivery in youtube: from traffic measurements to quality of experience”, in *Data Traffic Monitoring and Analysis: From measurement, classification and anomaly detection to Quality of experience*, M. M. Ernst Biersack Christian Callegari, Ed., Springer’s Computer Communications and Networks series, 2013.

[139] *ITU-R BT.500 Methodology for the Subjective Assessment of the Quality for Television Pictures*, Sep. 2009.

[140] *ITU-T P.910 Subjective Video Quality Assessment Methods for Multimedia Applications*, ITU-T, Apr. 2008.

[141] C. Keimel, J. Habigt, C. Horch, and K. Diepold, “Video Quality Evaluation in the Cloud”, in *Proceedings International Packet Video Workshop*, Munich, Germany, May 2012.

[142] K.-T. Chen, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, “A Crowdsourceable QoE Evaluation Framework for Multimedia Content”, in *Proceedings of the International Conference on Multimedia*, Beijing, China, Oct. 2009.

[143] C. Wu, K. Chen, Y. Chang, and C. Lei, “Crowdsourcing Multimedia QoE Evaluation: A Trusted Framework”, *Transactions on Multimedia*, vol. 15, no. 5, Jul. 2013.

[144] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, “Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea”, in *Proceedings of the Workshop on Quality of Multimedia Experience*, Yarra Valley, Australia, Jul. 2012.

[145] Q. Xu, Q. Huang, T. Jiang, B. Yan, W. Lin, and Y. Yao, “HodgeRank on Random Graphs for Subjective Video Quality Assessment”, *Transactions on Multimedia*, vol. 14, no. 3, Jun. 2012.

[146] C. Keimel, J. Habigt, and K. Diepold, “Challenges in Crowd-Based Video Quality Assessment”, in *Proceedings of the Workshop on Quality of Multimedia Experience*, Yarra Valey, Australia, Jul. 2012.

[147] R. Schatz, S. Egger, and K. Masuch, “The Impact of Test Duration on User Fatigue and Reliability of Subjective Quality Ratings”, *Journal of the Audio Engineering Society*, vol. 60, no. 1/2, Mar. 2012.

[148] T. Hoßfeld, R. Schatz, S. Biedermann, A. Platzer, S. Egger, and M. Fiedler, “The Memory Effect and Its Implications on Web QoE Modeling”, in *Proceedings of the International Teletraffic Congress*, San Francisco, USA, Sep. 2011.

[149] M. Sabou, K. Bontcheva, and A. Scharl, “Crowdsourcing Research Opportunities: Lessons from Natural Language Processing”, in *Proceedings of the Conference on Knowledge Management and Knowledge Technologies*, Graz, Austria, Sep. 2012.

[150] C. Harris, “You’re Hired! An Examination of Crowdsourcing Incentive Models in Human Resource Tasks”, in *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining*, Hong Kong, China, Feb. 2011.

[151] A. D. Shaw, J. J. Horton, and D. L. Chen, “Designing Incentives for Inexpert Human Raters”, in *Proceedings of the Conference on Computer Supported Cooperative Work*, Hangzhou, China, Mar. 2011.

[152] L. Von Ahn, “Games with a purpose”, *Computer*, vol. 39, no. 6, Jun. 2006.

[153] C. Eickhoff, C. G. Harris, A. P. de Vries, and P. Srinivasan, “Quality Through Flow and Immersion: Gamifying Crowdsourced Relevance Assessments”, in *Proceedings of Conference on Research and Development in Information Retrieval*, Portland, Oregon, USA, Aug. 2012.

[154] S. Owl, *Java Market Share*, Oct. 2012. [Online]. Available: http://www.statowl.com/java.php.

[155] B. Gardlo, M. Ries, T. Hoßfeld, and R. Schatz, "Microworkers vs. Facebook: The Impact of Crowdsourcing Platform Choice on Experimental Results", in *Proceedings of the Quality of Multimedia Experience*, Yarra Valley, Australia, Jul. 2012.

[156] *ITU-R BS.1116-1, Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems*, ITU Radiocommunication Assembly, 1997.

[157] M. Soleymani and M. Larson, "Crowdsourcing for affective annotation of video: Development of a viewer-reported boredom corpus", in *Proceedings of the Workshop on Crowdsourcing for Search Evaluation*, Geneva, Switzerland, Jul. 2010.

[158] O. Alonso, D. E. Rose, and B. Stewart, "Crowdsourcing for Relevance Evaluation", *SIGIR Forum*, vol. 42, no. 2, Nov. 2008.

[159] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor, "Are Your Participants Gaming the System?: Screening Mechanical Turk Workers", in *Proceedings of the Conference on Human Factors in Computing Systems*, Atlanta, Georgia, USA, Apr. 2010.

[160] T. Hoßfeld, R. Schatz, and S. Egger, "SOS: the MOS is not enough!", in *Proceedings of the Workshop on Quality of Multimedia Experience*, Mechelen, Belgium, Sep. 2011.

[161] S.-H. Kim, H. Yun, and J. S. Yi, "How to Filter out Random Clickers in a Crowdsourcing-Based Study?", in *Proceedings of the BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, Seattle, WA, USA, Oct. 2012.

[162] B. Gardlo, M. Ries, and T. Hoßfeld, "Impact of Screening Technique on Crowdsourcing QoE Assessments", in *Proceedings of the Radioelektronika Confernce*, Brno, Czech Republic, Apr. 2012.

[163] P.-Y. Hsueh, P. Melville, and V. Sindhwani, "Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria", in *Proceedings of the Workshop on Active Learning for Natural Language Processing*, Boulder, Colorado, Jun. 2009.

ISSN 1432-8801